

# BACHELORARBEIT

---

Zukunftssichere mobile Webseitenentwicklung  
auf Basis aktueller Webstandards

---

VON PHILIPP APLER

2012





# BACHELORARBEIT

---

Zukunftssichere mobile Webseitenentwicklung  
auf Basis aktueller Webstandards

---

AUTOR:	Philipp Apler
STUDIENGANG:	Medientechnik
SEMINARGRUPPE:	MT07w2-B

---

ERSTPFÜFER:	Prof. Dr.-Ing. Robert J. Wierzbicki
ZWEITPRÜFER:	Dipl.-Ing. Sieglinde Klimant

---

EINREICHUNG:	Mittweida, 31.10.2012
--------------	-----------------------



FACULTY OF MEDIA

---

# BACHELOR THESIS

---

Future-proof mobile web development based on current  
web standards

---

AUTHOR:	Philipp Apler
COURSE OF STUDIES:	Media Technology
SEMINAR GROUP:	MT07w2-B

---

FIRST EXAMINER:	Prof. Dr.-Ing. Robert J. Wierzbicki
SECOND EXAMINER:	Dipl.-Ing. Sieglinde Klimant

---

SUBMISSION:	Mittweida, 31.10.2012
-------------	-----------------------



## Bibliografische Angaben

Nachname, Vorname: Apler, Philipp

Thema der Arbeit: Zukunftssichere mobile Webseitenentwicklung  
auf Basis aktueller Webstandards

Topic of thesis: Future-proof mobile web development based on  
current web standards

2012 - 67 Seiten

Mittweida, Hochschule Mittweida (FH), University of Applied Sciences,  
Fakultät Medien, Bachelorarbeit, 2012

## Abstract

Diese Bachelorarbeit untersucht moderne Technologien und Methoden in der Webseitenentwicklung, wobei der Fokus auf der Realisierung plattformunabhängiger Webseiten liegt. Die Arbeit beginnt dabei mit einer Markt- und Verhaltensanalyse von modernen internetfähigen Endgeräten wie Smartphones und Tablet-PCs und zeigt, dass eine Optimierung auf mobile Endgeräte mittlerweile ein wichtiger Faktor für den Erfolg einer Webseite ist und nicht mehr ignoriert werden kann. Im Anschluss werden die modernen Webstandards HTML5 und CSS3 untersucht, sowie nützliche und innovative Einsatzmöglichkeiten von JavaScript vorgestellt. Der letzte Teil beschreibt ausführlich das Konzept Responsive Webdesign zur Erstellung plattformunabhängiger Webseiten und vergleicht die Methode mit der Alternative einer separaten mobilen Webseitenversion, wobei beide Ansätze auf ihre Vor- und Nachteile untersucht werden.

# Inhaltsverzeichnis

<b>Abstract</b>	<b>IV</b>
<b>Abbildungssverzeichnis</b>	<b>VII</b>
<b>Tabellensverzeichnis</b>	<b>IX</b>
<b>Abkürzungsverzeichnis</b>	<b>XI</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Webentwicklung im Zeitalter mobiler Endgeräte	1
1.2 Zielstellung und Gliederung der Arbeit	3
<b>2 Auf neuen Wegen ins World Wide Web</b>	<b>5</b>
2.1 Smartphones	5
2.1.1 Geschichtliche Entwicklung	6
2.1.2 Marktanalyse	8
2.2 Tablets	9
2.1.1 Geschichtliche Entwicklung	10
2.1.2 Marktanalyse	11
2.3 Nutzungsverhalten von Smartphone- und Tabletbesitzern	12
2.4 Unterscheidungsmerkmale der Endgeräte in Hinblick auf plattformübergreifene Webentwicklung	15
<b>3 Aktuelle Webstandards</b>	<b>18</b>
3.1 HTML5	18
3.1.1 Dokumententyp und Semantik	19
3.1.2 Das <canvas>-Element	21
3.1.3 Multimedia-Elemente	23
3.1.4 Das SVG-Format	24
3.1.5 Weitere Features von HTML5	25
3.2 CSS3	27
3.2.1 Rahmen	28



3.2.2	Farben und Farbverläufe	32
3.2.3	Schlagschatten	35
3.2.4	Multiple Hintergründe	37
3.2.5	Schriftarten	38
3.2.6	Selektoren	40
3.2.7	Transformationen	41
3.2.8	Ausblick & Fazit zu CSS3	43
3.3	Neue Einsatzmöglichkeiten von Javascript	48
3.3.1	JavaScript Bibliotheken	48
3.3.2	CSS-Extensions	49
<b>4</b>	<b>Responsive Webdesign</b>	<b>51</b>
4.1	Definition und Konzept	51
4.2	Analyse eines adaptiven Layouts	52
4.3	Die Alternative: Separate mobile Webseiten	56
4.4	Konzeption responsiver Webseiten	57
4.5	Technische Umsetzung	60
4.5.1	Media Queries	60
4.5.2	Viewport	62
4.5.3	Realisierung flexibler Breiten	62
4.5.4	Umsetzung einer Navigation	64
4.5.6	Gestaltung des Inhalts	67
4.6	Responsive Webdesign vs. separate mobile Webseite	67
<b>5</b>	<b>Zusammenfassung und Fazit der Arbeit</b>	<b>70</b>
	<b>Glossar</b>	<b>XII</b>
	<b>Quellenverzeichnis</b>	<b>XIV</b>
	<b>Eigenständigkeitserklärung</b>	<b>XXIV</b>

# Abbildungsverzeichnis

Abbildung 1	Übersicht internetfähige Medien, Seite 1
Abbildung 2	<i>Simon</i> von IBM, 1993, Quelle: <a href="http://research.microsoft.com">http://research.microsoft.com</a> , Seite 6
Abbildung 3	<i>Communicator</i> -Reihe von Nokia, ab 1996, Quelle: <a href="http://de.wikipedia.org">http://de.wikipedia.org</a> , Seite 6
Abbildung 4	<i>iPhone 1</i> von Apple, 2007 sowie das <i>T-Mobile G1</i> mit Android-Betriebssystem, Quellen: <a href="http://www.techfresh.net">http://www.techfresh.net</a> , <a href="http://www.amazon.de">http://www.amazon.de</a> , Seite 7
Abbildung 5	Vergleich Betriebssysteme <i>iOS</i> vs <i>Android</i> , Seite 8
Abbildung 6	Moderne Smartphones, v.l. n. r.: <i>Samsung Galaxy SII</i> , <i>iPhone 4</i> , <i>HTC Sensation</i> , <i>Nokia Lumia</i> , <i>Google Nexus</i> , <i>Motorola Pro+</i> , Quelle: <a href="http://amazon.de">http://amazon.de</a> , Seite 9
Abbildung 7	<i>Dynabook</i> 1968, Quelle: <a href="http://history-computer.com">http://history-computer.com</a> , Seite 10
Abbildung 8	<i>iPad 1</i> von Apple, 2000, Quelle: <a href="http://www.apple.com">http://www.apple.com</a> , Seite 10
Abbildung 9	Gegenüberstellung der verschiedenen Bildschirmauflösungen, Seite 15
Abbildung 10	Wischtechnik auf Touchscreens, angewandt an einer Web-Slideshow, Seite 17
Abbildung 11	Rechteck auf einem <code>&lt;canvas&gt;</code> -Element, jeweils 50 Pixel Abstand zum Rand, Seite 22
Abbildung 12	Skalierungsunterschiede zwischen JPG- und SVG- Format, Seite 24
Abbildung 13	Veranschaulichung der Drag&Drop-Funktionsweise, Seite 25
Abbildung 14	Einsatz abgerundeter Ecken auf <i>Youtube</i> und <i>Twitter</i> , Seite 28
Abbildung 15	Veranschaulichung der vorangehenden Beispiele <code>#beispiel-1</code> , <code>#beispiel-2</code> und <code>#beispiel-3</code> , Seite 30
Abbildung 16	Beispiel für <i>border-image</i> , Quelle: <a href="http://www.w3.org">www.w3.org</a> , Seite 31
Abbildung 17	Veranschaulichung Farbverläufe anhand <code>#beispiel-1</code> , <code>#beispiel-2</code> und <code>#beispiel-3</code> , Seite 34
Abbildung 18	Veranschaulichung Schatten anhand <code>#beispiel-1</code> und <code>#beispiel-2</code> , Seite 35

---

Abbildung 19	Veranschaulichung Textschatten anhand <i>#beispiel-1</i> und <i>#beispiel-2</i> , Seite 36
Abbildung 20	Veranschaulichung multipler Hintergrundbilder für <i>#multiple-bg</i> , Seite 38
Abbildung 21	Veranschaulichung der obigen Beispiele zu 2D Transformation, Seite 43
Abbildung 22	Veranschaulichung der im obigen Beispiel beschriebenen CSS-Transition, Seite 46
Abbildung 23	Screenshot der <i>Boston Globe</i> -Seite auf einem großen Monitor, Seite 52
Abbildung 24	Screenshot der <i>Boston Globe</i> -Seite einem Tablet quer und hochkant, sowie auf dem Smartphone, Seite 53
Abbildung 25	Layoutskizzierung der <i>Boston Globe</i> -Seite, Seite 53
Abbildung 26	Veranschaulichung des Layoutverhaltens bei kleiner werdender Browserbreite, Seite 54
Abbildung 27	Veranschaulichung des Layoutverhaltens bei noch kleiner werdender Browserbreite, Seite 55
Abbildung 28	Reduktion der Navigation, Seite 55
Abbildung 29	Beispiel für ein Akkordeon Element zum Ausklappen von Inhalten, sowie eine Bilderslideshow, Seite 58
Abbildung 30	Vergleich Beispiel-Navigation Desktop-Monitor vs. Smartphone Monitor, Seite 66

## Tabellenverzeichnis

Tabelle 1	Verkäufe Smartphones im Vergleich zu Computern weltweit im Jahr 2011, Quelle: <i>Canalys</i> , <a href="http://www.canalys.com/newsroom/smart-phones-overtake-client-pcs-2011">http://www.canalys.com/newsroom/smart-phones-overtake-client-pcs-2011</a> (29.10.2012), Seite 8
Tabelle 2	Weltweite Tablet Verkäufe - Vergleich 2011-2012, Quelle: <i>IDC</i> , <a href="http://www.idc.com/getdoc.jsp?containerId=prUS23632512">http://www.idc.com/getdoc.jsp?containerId=prUS23632512</a> (29.10.2012), Seite 11
Tabelle 3	Nutzung des internetfähigen Mobilgeräts, Quelle: <i>AGOF Mobile Facts 2011</i> (Seite 24), Seite 13
Tabelle 4	App vs. mobiler Browser, Quelle: <i>APPRUPT</i> Studie zur App-Nutzung und Kriterien für den App-Download (Seite 2), Seite 14
Tabelle 5	Browsersupport <canvas>-Element, ab welcher Version, Quelle: <a href="http://caniuse.com">http://caniuse.com</a> , Seite 21
Tabelle 6	Ab welcher Version wird SVG-Format von Browsern unterstützt, Quelle: <a href="http://caniuse.com">http://caniuse.com</a> , Seite 25
Tabelle 7	Ab welcher Version wird <i>border-radius</i> von Browsern unterstützt, Quelle: <a href="http://caniuse.com">http://caniuse.com</a> , Seite 30
Tabelle 8	Ab welcher Version wird <i>border-image</i> von Browsern unterstützt, Quelle: <a href="http://caniuse.com">http://caniuse.com</a> , Seite 32
Tabelle 9	Ab welcher Version wird das HSL-Format sowie Transparenz von Browsern unterstützt, Quelle: <a href="http://caniuse.com">http://caniuse.com</a> , Seite 33
Tabelle 10	Ab welcher Version werden Farbverläufe von Browsern unterstützt, Quelle: <a href="http://caniuse.com">http://caniuse.com</a> , Seite 34
Tabelle 11	Ab welcher Version werden Schatten von Browsern unterstützt, Quelle: <a href="http://caniuse.com">http://caniuse.com</a> , Seite 37
Tabelle 12	Ab welcher Version werden multiple Hintergrundbilder von Browsern unterstützt, Quelle: <a href="http://caniuse.com">http://caniuse.com</a> , Seite 38
Tabelle 13	Übersicht Browsersupport der Font-Formate, Quelle: <a href="http://socialcompare.com/en/comparison/browser-fonts-support-comparison">http://socialcompare.com/en/comparison/browser-fonts-support-comparison</a> , Seite 39
Tabelle 14	Ab welcher Version wird das <i>@font-face</i> -Attribut von Browsern verstanden, Quelle: <a href="http://caniuse.com">http://caniuse.com</a> , Seite 40

Tabelle 15	Ab welcher Version unterstützen Browser CSS3 Selektoren, Quelle: <a href="http://caniuse.com">http://caniuse.com</a> , Seite 41
Tabelle 16	Ab welcher Browserversion werden 2D Transformationen unterstützt, Quelle: <a href="http://caniuse.com">http://caniuse.com</a> , Seite 43
Tabelle 17	Ab welcher Browserversion werden 3D Transformationen unterstützt, Quelle: <a href="http://caniuse.com">http://caniuse.com</a> , Seite 44
Tabelle 18	Ab welcher Version unterstützen Browser CSS3 Transitions, Quelle: <a href="http://caniuse.com">http://caniuse.com</a> , Seite 46
Tabelle 19	Ab welcher Version wird ein <i>multiple-column</i> -Layout von Browsersn unterstützt, Quelle: <a href="http://caniuse.com">http://caniuse.com</a> , Seite 47
Tabelle 20	Ab welcher Version wird die <i>Flexbox</i> von Browsersn unterstützt, Quelle: <a href="http://caniuse.com">http://caniuse.com</a> , Seite 47

# Abkürzungsverzeichnis

API	Application Programming Interface
CSS	Cascading Style Sheets
EOT	Embedded OpenType, Fontformat
HTML	Hypertext Markup Language
IBM	International Business Machines
ID	Selektor für CSS
JPG	Joint Photographic Expert Group
KB	KiloByte
MP3	MPEG-1 Audio Layer III
MP4	MPEG-1 Audio Layer IV
MPEG	Moving Picture Experts Group
OGG	ogg Vorbis - Audioformat
PNG	Portable Network Graphics
px	Pixel
SVG	Scalable Vector Graphics
SVGZ	komprimierte Scalable Vector Graphic
TTF	TrueType Font
URL	Uniform Resource Locator
WAV	Waveform Audio File Format (WAVE)
W3C	World Wide Web Consortium
WebM	Video-Containerformat
WHATWG	Web Hypertext Application Technology Working Group
WOFF	Web Open Font Format
XHTML	Extensible HyperText Markup Language
XML	Extensible Markup Language



# 1 Einleitung

## 1.1 Webentwicklung im Zeitalter mobiler Endgeräte

In den vergangenen Jahren wurde die multimediale Landschaft durch den Siegeszug von Smartphone und Tablet zunehmend vielfältiger und komplexer. Eine der Folgen dieser Entwicklung ist, dass PC und Laptop längst nicht mehr die einzigen internetfähigen Medien von Relevanz sind. Gesurft wird nun auch unterwegs mit dem Smartphone und von zu Hause aus via Tablet, Kindle oder Netbook. Selbst einige aktuelle Spielekonsolen haben integrierte Webbrowser<sup>1</sup> und moderne Fernsehgeräte können mittels Smart-TV auf das Internet zugreifen.<sup>2</sup>

Für das World Wide Web hat die zunehmende Diversität der Endgeräte weitreichende Konsequenzen. Bis heute ist es bevorzugte Praxis, Webseiten auf Basis eines statischen Layouts mit fixen Höhen und Breiten zu entwickeln. Die Maße sind dabei auf die gängigen Bildschirmauflösungen optimiert. Lagen diese vor einigen Jahren noch bei 800 x 600 oder 1024 x 768 Pixeln, wurden später auch Monitore mit einer Auflösung von 1280 x 1024 Pixeln immer gebräuchlicher.<sup>3</sup>

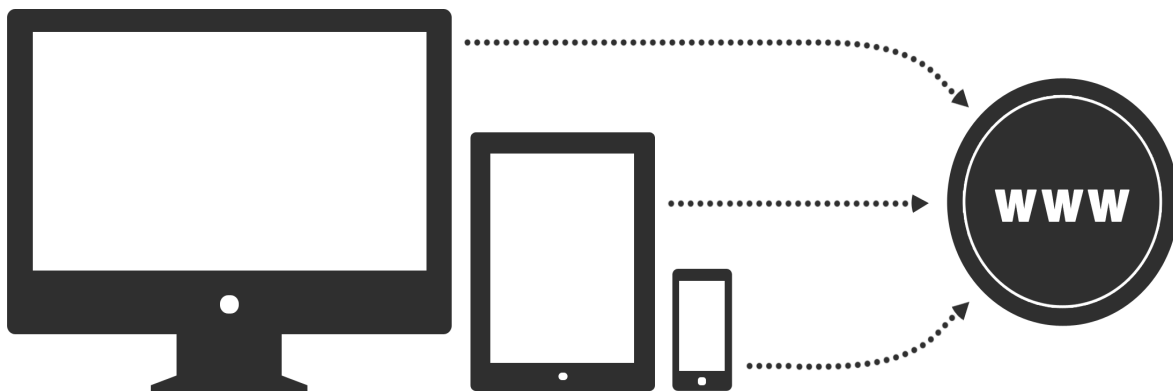


Abb. 1: Übersicht internetfähige Medien

- 
- 1 Vgl. Nintendo: Nintendo DS Browser, [http://www.nintendo.de/NOE/de\\_DE/games/nds/nintendo\\_ds\\_browser\\_3151.html](http://www.nintendo.de/NOE/de_DE/games/nds/nintendo_ds_browser_3151.html) (29.10.2012)
  - 2 Vgl. Samsung: Anwendungsmöglichkeiten von Samsung Smart TV [http://de.samsung.com/de/microsites/smarttv/anwendungen\\_feature.aspx](http://de.samsung.com/de/microsites/smarttv/anwendungen_feature.aspx) (29.10.2012)
  - 3 Vgl. Webhits: Web-Barometer, <http://www.webhits.de/deutsch/index.shtml?webstats.html> (29.10.2012)



Heute sieht die Situation jedoch vollkommen anders aus. Im Zeitalter von Smartphone und Tablet wird von faustgroßen Bildschirmen aus bis hin zu Monitoren mit Auflösungen jenseits der 1920x1080 Pixel gleichermaßen im Internet gesurft. Angesichts dieser Mannigfaltigkeit erscheint das Konzept statischer Layouts nicht mehr sinnvoll. Die Benutzerfreundlichkeit leidet stark, wenn die Inhalte auf dem vergleichsweise kleinen Touchscreen eines Smartphones hin- und hergeschoben und die relevanten Bereiche mühsam herangezoomt werden müssen. Damit die Absprungrate<sup>4</sup> aufgrund mangelnder Zugänglichkeit nicht unnötig steigt, muss eine moderne und zukunftsichere Webseite das Ziel haben, Benutzern jedes Endgeräts ein gleichhohes Maß an Benutzerfreundlichkeit bieten zu können.

Auf welchem Weg dieses Ideal erreicht werden kann, ist die Frage, mit der sich die Webentwicklergemeinschaft derzeit umfassend auseinandersetzt. Dabei werden ständig neue Antworten gefunden, mit denen die Usability einer Webseite auf allen Endgeräten erhöht und endgerätspezifische Probleme gelöst werden. Als ein Ergebnis dieses Entwicklungsprozesses haben viele große und kleine Webportale mittlerweile eine separate mobile Webseitenversion eingerichtet. Werden diesen via Smartphone besucht, wird direkt auf die mobile Version umgeleitet, welche in ihrer Darstellung und Bedienung an das jeweilige Betrachtungsgerät angepasst wurde.

Auf den Grundlagen des sogenannten Responsive Webdesign erblicken nun auch immer mehr Webseiten das Licht des Internets, welche ein vollkommen neues Konzept verfolgen. Nach dem Leitsatz „Ein Design für alle Endgeräte“ zeichnen sich diese Seiten besonders durch ihre umfassende Anpassungsfähigkeit und ihr flexibles und dynamisches Grundgerüst aus.

Um diese neuartigen Konzepte jedoch auch technisch umsetzen zu können, arbeitet das *World Wide Web Consortium* (kurz W3C) durchweg an der Weiterentwicklung aktueller Webstandards wie HTML5 und CSS3. Diese bilden die Grundlage in der voranschreitenden Entwicklung plattformübergreifenden Webdesigns.

---

4 Die Absprungrate beschreibt alle Webseitenzugriffe, bei denen maximal eine Seite aufgerufen wurde.

## 1.2 Zielstellung und Gliederung der Arbeit

Diese Bachelorarbeit soll aufzeigen, wie zukunftsichere und plattformübergreifende Webseiten realisiert werden können. Neben der Vorstellung aktueller Webstandards wird der Schwerpunkt auf die Realisierung mobiler und responsiver Webseiten gelegt. Dabei geht es neben der Konzeption und den gestalterischen Aspekten auch um die technische Umsetzung von plattformunabhängigem Webdesign.

Das erste Kapitel beschreibt die derzeitigen Entwicklungen der multimedialen Gesellschaft und die Folgen für die Webentwicklung. Weiterhin wird die Zielstellung und Vorgehensweise dieser Arbeit erläutert.

Das zweite Kapitel stellt die verschiedenen modernen Endgeräte vor. Der Schwerpunkt liegt dabei auf Smartphones und Tablets. Eine umfassende Marktanalyse der Geräte wird die Notwendigkeit der plattformübergreifenden Webentwicklung aufzeigen. Außerdem soll das Nutzungsverhalten von Smartphone- und Tabletbesitzern bezüglich der Surfgewohnheiten untersucht werden. Abschließend werden die für die Webentwicklung relevanten Unterscheidungsmerkmale der Endgeräte herausgearbeitet.

Kapitel 3 erläutert die technischen Grundlagen für zukunftsichere Webseiten. Im Fokus steht der aktuelle Webstandard HTML5, insbesondere die für eine plattformübergreifende Webentwicklung relevanten Neuerungen und Möglichkeiten. Nachfolgend wird auf CSS3 und die damit verbundenen gestalterischen Möglichkeiten eingegangen sowie verschiedene neue Einsatzmöglichkeiten von JavaScript untersucht. Die Arbeit wird jedoch keine komplette Einführung in diese drei Webtechnologien liefern und setzt gewisse Grundkenntnisse voraus.

Das folgende Kapitel 4 bildet den Schwerpunkt der Arbeit. Ausführlich wird hier der Weg zu einer plattformübergreifenden Webseite unter konzeptionellen, gestalterischen und technischen Aspekten beschrieben. Untersucht werden deren Planung und Umsetzung. An verschiedenen Beispielen werden danach die Einsatzgebiete sowie Vor- und Nachteile von mobilen Webseiten analysiert. Außerdem wird sich speziell mit dem Konzept des Responsive Webdesign auseinandergesetzt. Die neuartige

Technik wird vorgestellt und die Funktionsweise responsiver Webseiten untersucht.

Im abschließenden Kapitel 5 wird die Arbeit zusammengefasst und ein Ausblick auf den weiteren Weg der mobilen Webentwicklung gegeben.

## 2 Auf neuen Wegen ins World Wide Web

Betrachtet man alle modernen internetfähigen Medien, haben vor allem zwei Geräteklassen das gesellschaftliche Leben in den letzten Jahren nachhaltig geprägt: Smartphones und Tablets. Über 23 Millionen Deutsche (Stand: März 2012) besitzen und nutzen ein Smartphone, mit steigender Tendenz.<sup>5</sup> Weltweit wurden im Jahr 2011 fast eine halbe Milliarde der multimedialen Geräte verkauft.<sup>6</sup> Und auch Tablet-Computer haben im letzten Jahr über 60 Millionen mal einen neuen Besitzer gefunden.<sup>7</sup> Nachfolgend sollen daher die Geräte genauer vorgestellt werden.

### 2.1 Smartphones

Um die Anfänge der Smartphones erkennen zu können, muss man deren Merkmale genauer definieren. So ist das wohl prägnanteste Merkmal die bereits geschilderte Multifunktionalität. Das Smartphone ist nicht wie das klassische Mobiltelefon auf das Telefonieren optimiert, sondern soll die komfortable Bedienung verschiedenster Anwendungen ermöglichen. So verwenden aktuelle Smartphones beispielsweise große, hochauflösende Bildschirme, alphanumerische Tastaturen oder auch Touchscreens.

Haben klassische Mobiltelefone meist eine vordefinierte Benutzeroberfläche, so besitzen heutige Smartphones ein Betriebssystem, welches sich frei konfigurieren lässt und die Installation weiterer Programme ermöglicht. Weiterhin besitzen die Geräte im Zuge der Multifunktionalität unterschiedliche Sensoren wie Lage-, Bewegungs- oder Lichtsensoren sowie einen GPS-Empfänger. Es lassen sich außerdem häufig die technischen Voraussetzungen zu gängigen Datenübertragungsarten wie WLAN, Bluetooth, Infrarot, sowie eine USB-Kabelverbindung in einem Smartphone finden.

---

5 Vgl. Statistika GmbH: Smartphone-Nutzer in Deutschland von 2009 bis 2012, <http://de.statista.com/statistik/daten/studie/198959/umfrage/anzahl-der-smartphonenuutzer-in-deutschland-seit-2010/> (29.10.2012)

6 Vgl. IDC: Top Five Worldwide Smartphone Vendors, 06.02.2012, <http://www.idc.com/getdoc.jsp?containerId=prUS23299912> (29.10.2012)

7 Vgl. Gartner: Worldwide Sales of Media Tablets to End Users by OS, 10.04.2012 <http://www.gartner.com/it/page.jsp?id=1980115> (29.10.2012)

## 2.1.1 Geschichtliche Entwicklung

Betrachtet man Smartphones als multifunktionale Telefone, lassen sich bereits Ende des 20. Jahrhunderts Mobiltelefone finden, auf die die Bezeichnung „Smartphone“ zutrifft. Einige wichtige Vertreter sollen nachfolgend kurz vorgestellt werden.

So verkaufte *IBM* im Jahr 1993 erstmal ein Mobiltelefon mit der Bezeichnung *Simon* (Abbildung 2), das gleichzeitig Pager, PDA und Faxgerät war.<sup>8</sup> Bedient wurde es über ein schwarz-weißes Touchscreen, wobei Programme über Icons geöffnet wurden. Es enthielt außerdem eine E-Mail-, Kalender- und Taschenrechnerfunktion und besaß einen ein Megabyte großen Speicher. Dieses und folgende Smartphone-Produkte waren jedoch für den Normalbürger kaum attraktiv. Gründe dafür waren die große, klobige Ziegelform, das hohe Gewicht und der enorme Preis - *Simon* kostete 899 Dollar. Diese ersten Smartphones waren nur auf die Verwendung in Unternehmen ausgelegt.



Abb. 2: *Simon* von *IBM*, 1993,  
Quelle: <http://research.microsoft.com>



Abb. 3: *Communicator*-Reihe von *Nokia*, ab 1996,  
Quelle: <http://de.wikipedia.org>

Weiterentwickelt wurde das Konzept Smartphone von *Nokia*. Mit der 1996 eingeführten *Communicator*-Reihe (Abbildung 3) platzierte das Unternehmen ein Produkt auf dem Markt, welches mit einer vollständigen ausklappbaren QWERTZ-Tastatur

---

8 Vgl. Bloomberg Businessweek: Before iPhone and Android Came Simon, 29.06.2012, <http://www.businessweek.com/articles/2012-06-29/before-iphone-and-android-came-simon-the-first-smartphone> (29.10.2012)

und einem TFT-Bildschirm ausgestattet war und sogar HTML-Seiten darstellen konnte.<sup>9</sup> Jedoch war auch hier wieder der hohe Preis der Grund dafür, dass die Geräte auf dem Markt nie richtig Fuß fassen konnten. Trotzdem wird *Nokia* in den Folgejahren Marktführer im Bereich Smartphones (2007: über 50 Prozent Marktanteil<sup>10</sup>).

Einen großen Schritt markierte 2007 *Apple's* Einstieg in den Smartphone-Markt. Mit dem ersten *iPhone* und dessen Erfolg wurden Smartphones vom Business- zum Massenprodukt. Es besaß eine neuartige intuitive Bedienoberfläche auf Basis eines Touchscreens und ein eigenes Betriebssystem (*iOS*), was erstmals um immer weitere Anwendungen erweitert werden konnte.<sup>11</sup> Der prozentuale Anteil an verkauften *iOS*-Smartphones stieg in den Folgejahren auf über 20 Prozent.<sup>12</sup> Ende 2008 wurde das erste Smartphone mit dem Betriebssystem *Android* verkauft.<sup>13</sup>



Abb. 4: *iPhone 1* von *Apple*, 2007 sowie das *T-Mobile G1* mit *Android*-Betriebssystem,  
Quellen: <http://www.techfresh.net>, <http://www.amazon.de>

- 
- 9 Vgl. [nokiamuseum.com](http://nokiamuseum.com): Nokia 9000,  
<http://nokiamuseum.com/view.php?model=9000> (29.10.2012)
- 10 Vgl. Statistika GmbH - Global market share of Nokia smartphones since 2012,  
<http://www.statista.com/statistics/12861/market-share-of-nokia-smartphones-since-2007/>  
(29.10.2012)
- 11 Vgl. Wilson, Tracy V.: How the iPhone Works,  
<http://electronics.howstuffworks.com/iphone5.htm> (29.10.2012)
- 12 Vgl. Gartner: World-Wide Smartphone Sales,  
[http://en.wikipedia.org/wiki/File:World\\_Wide\\_Smartphone\\_Sales\\_Share.png](http://en.wikipedia.org/wiki/File:World_Wide_Smartphone_Sales_Share.png) (29.10.2012)
- 13 Vgl. HTC: T-Mobile Unveils the T-Mobile G1 - the First Phone Powered by Android, 23.09.2012,  
<http://web.archive.org/web/20110712230204/http://www.htc.com/www/press.aspx?id=66338&lang=1033> (29.10.2012)



## 2.1.2 Marktanalyse

Der Smartphone-Markt boomt. Wurden 2011 noch insgesamt über 487 Millionen Exemplare verkauft<sup>13</sup>, erreicht man diese Verkaufszahlen nun beinahe schon innerhalb eines Quartals<sup>14</sup>. Damit liegt der Verkauf von Smartphones längst höher als der von Notebooks und Desktop-PCs (Tabelle 1).

	SMARTPHONES	COMPUTER GESAMT	TABLETS	NETBOOKS	NOTEBOOKS	DESKTOPS
VERKÄUFE 4. QUARTAL 2011 (IN MILLIONEN)	158,5	120,2	26,2	6,7	57,9	29,1
VERKÄUFE GESAMT 2011 (IN MILLIONEN)	487,7	414,6	63,2	29,4	209,6	112,4

Tabelle 1: Verkäufe Smartphones im Vergleich zu Computern weltweit im Jahr 2011,

Quelle: *Canalys*, <http://www.canalys.com/newsroom/smart-phones-overtake-client-pcs-2011> (29.10.2012)

Bei den Herstellern liegt *Apple* mit 29 Prozent Marktanteil im ersten Quartal 2012 noch knapp vor *Samsung* mit 24 Prozent.<sup>15</sup> Betrachtet man jedoch die Betriebssysteme, so hat *Apple* seine Marktführung längst eingebüßt. Das offene *Android*-Betriebssystem, welches von den meisten Konkurrenzprodukten unterschiedlichster Hersteller verwendet wird, liegt mit einem Marktanteil von 46,9 Prozent weit vor den 18,8 Prozent des *iOS*. Gefolgt wird *Apple* von *Nokias Symbian* mit 16,9 Prozent.<sup>16</sup>



Abb. 5: Vergleich Betriebssysteme *iOS* vs *Android*

<sup>13</sup> Vgl. *Canalys*: Worldwide smart phone market, full year 2011, 03.02.2012, <http://www.canalys.com/newsroom/smart-phones-overtake-client-pcs-2011> (29.10.2012)

<sup>14</sup> Vgl. *Gartner*: Worldwide Mobile Terminal Sales to End Users in 1Q11, 19.05.2011, <http://www.gartner.com/it/page.jsp?id=1689814> (29.10.2012)

<sup>15</sup> Vgl. *NPD Group*: Top U.S. Smartphone Manufacturers, 02.05.2012, [https://www.npd.com/wps/portal/npd/us/news/press-releases/pr\\_120502/](https://www.npd.com/wps/portal/npd/us/news/press-releases/pr_120502/) (29.10.2012)

<sup>16</sup> Vgl. *IDC*: Top Smartphone Operating Systems, 08.08.2012, <http://www.idc.com/getdoc.jsp?containerId=prUS23638712> (29.10.2012)

In Deutschland wurden im Jahr 2011 rund 11,8 Millionen Smartphones verkauft. Somit handelte es sich bei 43 Prozent aller verkauften Mobiltelefone um ein Smartphone. Der Absatz nahm im Vergleich zum vergangenen Jahr um 31 Prozent zu. Laut den Daten des Bundesverbandes Informationswirtschaft, Telekommunikation und neue Medien e.V. (BITKOM) wird der Smartphone-Absatz den der normalen Handys im Jahr 2012 in Deutschland erstmals überholen.<sup>17</sup>



Abb. 6: Moderne Smartphones, v. l. n. r.: Samsung Galaxy SII, iPhone 4, HTC Sensation, Nokia Lumia, Google Nexus, Motorola Pro+, Quelle: <http://amazon.de>

## 2.2 Tablets

Neben Smartphones haben sich vor allem Tablet-Computer als moderner Begleiter in unserer Gesellschaft etabliert. Tablets existieren mittlerweile in allen Farben und Formen, doch es gibt einige Spezifikationen, die sie alle gemeinsam haben. So erkennt man die kleinen, mobilen Computer immer an der flachen Form, ähnlich einer Schreibtafel für Kinder. Auch die Größe bewegt sich immer in einer Spanne zwischen Smartphones und Laptop. Etabliert hat sich die Verwendung eines Touchscreens und einer Bedienung der Geräte per Hand. Es existieren jedoch auch Tablets, die für eine Steuerung via Stift ausgelegt sind.<sup>18</sup> Heutige Tablet-Computer sind drahtlos, batteriebetrieben und zeichnen sich durch vielfältige technische Möglichkeiten aus. Der Nutzungsfokus der Geräte liegt dabei auf multimedialen Tätigkeiten wie dem Lesen von Büchern, dem Anschauen von Filmen sowie dem Surfen im Internet.

17 Vgl. BITKOM: Smartphone-Absatz steigt rasant, 09.01.2012, [http://www.bitkom.org/de/markt\\_statistik/64086\\_70921.aspx](http://www.bitkom.org/de/markt_statistik/64086_70921.aspx) (29.10.2012)

18 Vgl. Samsung: Samsung Galaxy Note 10.1, <http://www.samsung.com/de/consumer/mobile-device/galaxy-note/galaxy-note/GT-N8010ZWADBT> (29.10.2012)



## 2.2.1 Geschichtliche Entwicklung

Die Geschichte des Tablets beginnt nicht erst mit dem Siegeszug des iPad im Jahr 2010, das Konzept des drahtlosen, flachen Multimedia-Computer ist bereits mehr als 40 Jahre alt. 1968 entwickelte Alan Kay das *Dynabook* (Abbildung 7), ein flacher Laptop, der vor allem für Kinder gedacht war.<sup>19</sup> 1993 beginnt die Firma *Apple*, sich mit ersten Tablet-Computern auf dem Markt zu positionieren. Die Produktreihe der *Newton MessagePads* hatte allerdings nur geringen Erfolg.<sup>20</sup> In den Folgejahren wurden Tablet-Computer als Personal Digital Assistant (PDA) bezeichnet und dienten hauptsächlich der Kalender-, Adressen- und Aufgabenverwaltung. Sie besaßen keinen Internetzugang.

Erste internetfähige Tablets erscheinen beispielsweise im Jahr 2000 mit dem *Microsoft Tablet PC*, jedoch ist keines der Modelle besonders erfolgreich.<sup>21</sup> Erst das 2010 vorgestellte *iPad* von *Apple* kann sich erfolgreich auf dem Markt platzieren<sup>22</sup> und schließt mit seinen funktionellen Möglichkeiten die Lücke zwischen Laptop und PDA. Mit für an Touchscreens angepassten Programmen, Internetzugang und leistungsfähiger Hardware leitet das *iPad* den Siegeszug der Tablets ein (Vgl. Punkt 2.2.2 Marktanalyse).

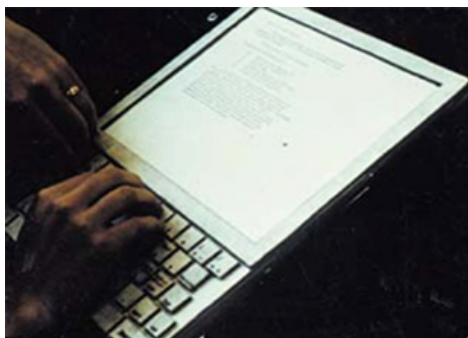


Abb. 7: *Dynabook* 1968,  
Quelle: <http://history-computer.com>



Abb. 8: *iPad 1* von *Apple*, 2000, Quelle: <http://www.apple.com>

- 
- 19 Vgl. Computer History Museum: The 40th Anniversary of the Dynabook, <http://www.computerhistory.org/collections/accession/102695024> (29.10.2012)
- 20 Vgl. GIZMODO: The Story Behind Apple's Newton, 19.01.2010, <http://gizmodo.com/5452193/the-story-behind-apples-newton> (29.10.2012)
- 21 Vgl. channelpartner.de: Absatzzahlen von Tablet-PCs weiterhin ein Trauerspiel, 10.11.2003, <http://www.channelpartner.de/news/209592/index.html> (29.10.2012)
- 22 Vgl. Gartner: Worldwide Media Tablet Sales 2010, 15.10.2010, <http://www.gartner.com/it/page.jsp?id=1452614> (29.10.2012)

## 2.2.2 Marktanalyse

Noch haben sich Tablets nicht so umfassend in unseren Alltag integriert wie es bei Smartphones bereits der Fall ist. Trotzdem werden laut dem Marktforschungsinstitut *International Data Corporation (IDC)* bis Ende 2012 weltweit über 117 Millionen Exemplare verkauft - mit steigender Tendenz.<sup>23</sup> Für das Jahr 2013 werden sogar über 165 Millionen Verkäufe vorausgesagt. Dominiert wird der Markt dabei immer noch von der Firma *Apple*, welche im zweiten Quartal des Jahres 2012 einen Marktanteil von 68,2 Prozent besaß und damit fast so viele Tablets verkaufte wie im gesamten Jahr 2010.<sup>24</sup> Im Vergleich zum Vorjahr hat sich *Apple* hier sogar noch steigern können. Größte Konkurrenten sind *Samsung* mit seinen *Galaxy Tab* Geräten und *Amazon* mit seinen *Kindle* E-Readern. Beide haben jedoch bisher nie einen Marktanteil von über zehn Prozent erreichen können.<sup>25</sup>

FIRMA	2. QUARTAL 2012 VERKÄUFE (IN TAUSEND)	MARKTANTEIL	2. QUARTAL 2011 VERKÄUFE (IN TAUSEND)	MARKTANTEIL	WACHSTUM 2011 ZU 2012
1. Apple	17.042	68,2%	9.248	61,5%	84,3%
2. Samsung	2.391	9,6%	1.099	7,3%	117,6%
3. Amazon.com	1.252	5,0%	0	-	-
4. ASUS	855	3,4%	397	2,6%	115,5%
5. Acer	385	1,5%	629	4,2%	-38,7%
Andere	3.067	12,3%	3.668	24,4%	-16,4%
<b>Gesamt</b>	<b>24.994</b>	<b>100%</b>	<b>15.042</b>	<b>100%</b>	<b>66,2%</b>

Tabelle 2: Weltweite Tablet Verkäufe - Vergleich 2011-2012, Quelle: IDC, <http://www.idc.com/getdoc.jsp?containerId=prUS23632512> (29.10.2012)

Wie auch bei den Smartphones sieht die Situation anders aus, wenn man auf die Marktverteilung der Betriebssysteme von Tablets schaut. Denn auch hier nutzen eine Vielzahl der Geräte verschiedenster Hersteller das Betriebssystem *Android*. Selbst die *Kindle*-Reihe von *Amazon* benutzt das *Android*-Betriebssystem in einer überarbeite-

23 Vgl. IDC: Worldwide Media Tablet Shipments, Second Quarter 2012, 02.08.2012, <http://www.idc.com/getdoc.jsp?containerId=prUS23632512> (29.10.2012)

24 Vgl. Gartner: Worldwide Media Tablet Sales to End Users, 15.10.2011, <http://www.gartner.com/it/page.jsp?id=1452614> (29.10.2012)

25 Vgl. IDC: Worldwide Media Tablet Shipments, Second Quarter 2012, 02.08.2012, <http://www.idc.com/getdoc.jsp?containerId=prUS23632512> (29.10.2012)

ten, angepassten Version.<sup>26</sup> Laut einer Studie des *Pew Research Center's for Excellence in Journalism* besitzen fast die 50 Prozent der Tablet-Besitzer ein *Android*-Gerät<sup>27</sup>, davon sind fast die Hälfte *Kindle*-Tablets.

## 2.3 Nutzungsverhalten von Smartphone- und Tabletbesitzern

Die Marktanalyse hat bestätigt, dass Smartphones ein fester Bestandteil unserer Gesellschaft geworden sind. Die für diese Arbeit relevante Frage ist nun, wie die Geräte letztendlich eingesetzt und genutzt werden.

Zwar ist das Smartphone immer noch hauptsächlich ein Telefonier-Medium, jedoch nutzen laut einer Umfrage der *Statistika GmbH* 65 Prozent das Gerät täglich, um im Internet zu surfen. Damit es es neben der Verwaltung von E-Mails und Adressbuch die zweithäufigste Tätigkeit mit einem Smartphone.<sup>28</sup> Vergleicht man die Studie *mobile facts* der *Arbeitsgemeinschaft Online Forschung (AGOF)* aus dem Jahr 2010 mit den aktuellen Ergebnissen von 2011, so ist die Anzahl der Nutzer von mobilem Internet um 55 Prozent gewachsen. Es war ein Anstieg von 10,95 Millionen auf 16,95 Millionen Nutzer zu verzeichnen.<sup>29</sup> Entsprechend der Erhebungsmethode der *AGOF*-Studie bedeutet das, dass innerhalb des 30-tägigen Erhebungszeitraums 24,1 Prozent der deutschsprachigen Bevölkerung auf eine mobile Webseite oder eine mobile Application zugegriffen haben.

Die naheliegende Frage ist, wie diese Personen das mobile Internet mit ihrem Smartphone und Tablets nun genau nutzen. Grundlegend ist festzustellen, dass das stationäre

---

26 Vgl. Amazon: Amazon Kindle technische Details,  
[http://www.amazon.de/Kindle-Zoll-Ink-Display-WLAN-Schwarz/dp/B007HCCOD0/ref=sr\\_tr\\_sr\\_1?ie=UTF8&qid=1351501107&sr=8-1](http://www.amazon.de/Kindle-Zoll-Ink-Display-WLAN-Schwarz/dp/B007HCCOD0/ref=sr_tr_sr_1?ie=UTF8&qid=1351501107&sr=8-1) (29.10.2012)

27 Vgl. Pew Research Center's for Excellence in Journalism: The Explosion in Mobile Audiences And A Close Look At What It Means For News, 01.10.2012,  
[http://www.journalism.org/analysis\\_report/future\\_mobile\\_news](http://www.journalism.org/analysis_report/future_mobile_news) (29.10.2012)

28 Vgl. Statistika GmbH: Smartphone-Funktionen - Nutzung 2012,  
<http://de.statista.com/statistik/daten/studie/166150/umfrage/nutzung-von-smartphone-funktionen-in-deutschland/> (29.10.2012)

29 Vgl. AGOF: Mobile Facts 2011, Seite 14

Internet bis jetzt nicht durch das Mobile ersetzt wurde. Es findet kein Substitutionsprozess statt, 70,1 Prozent nutzen das mobile Internet eher zusätzlich zum stationären, 15,3 Prozent nutzen beides gleichwertig. Lediglich für 14,6 Prozent spielt das mobile Internet bereits eine wichtigere Rolle als das Stationäre.<sup>30</sup>

„Ich nutze das mobile Internet mindestens 1x Pro Woche für ...“

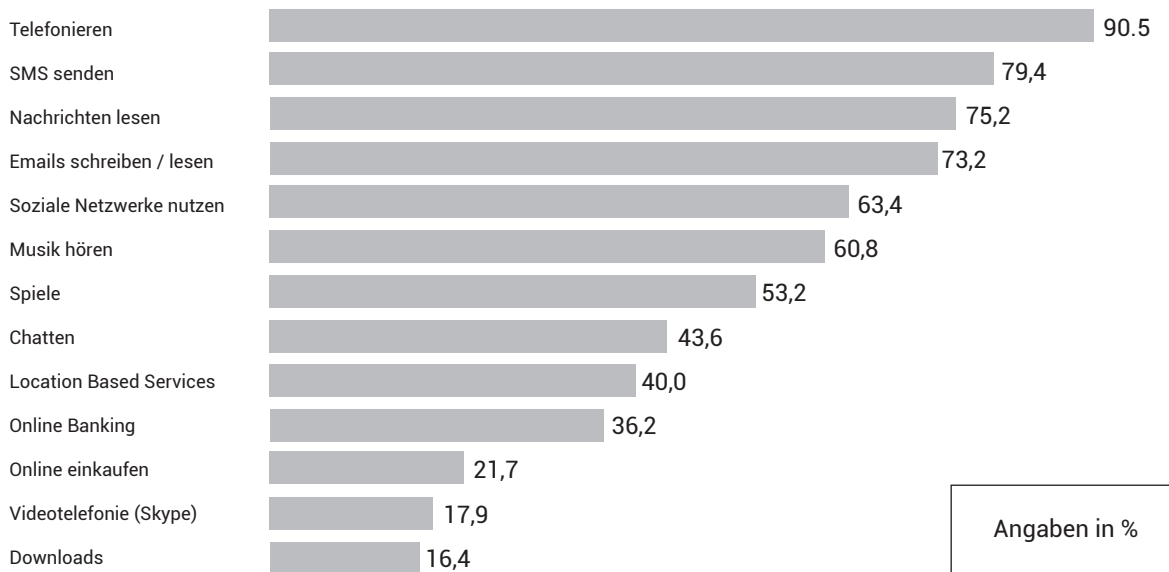


Tabelle 3: Nutzung des internetfähigen Mobilgeräts, Quelle: AGOF Mobile Facts 2011, Seite 24

Deutlich wird, dass neben den klassischen Verwendungszwecken wie Telefonieren und SMS versenden vor allem das Schreiben und Lesen von Emails sowie die Nutzung sozialer Netzwerke von Bedeutung ist. Jene Tätigkeiten wären ohne das mobile Internet nicht möglich und zeigen daher, welchen Stellenwert dieses bereits innehat. Auch der Bereich des Online-Bankings wird von über einem Drittel der Befragten bereits wöchentlich mithilfe von Smartphone und mobilem Internet gehandhabt. Außerdem kauft bereits jeder fünfte mobile Internetnutzer einmal pro Woche online ein.

Viele Aktivitäten kann man sowohl über den Browser als auch mittels einer Application ausführen. Dabei haben sich für bestimmte Tätigkeiten eine der beiden Methoden als vorteilhaft erwiesen. Für komplexere Anwendungsgebiete wie Spiele oder Navigation

30 Vgl. AGOF: Mobile Facts 2011, Seite 23

kommen überwiegend Apps zum Einsatz, wohingegen für einfachere Online-Dienste wie das Buchen von Fahrkarten, Einkaufen oder Online-Banking der mobile Browser zum Einsatz kommt.<sup>31</sup> Hier entscheidet letztendlich, welche Form der Nutzung komfortabler ist sowie die Notwendigkeit.

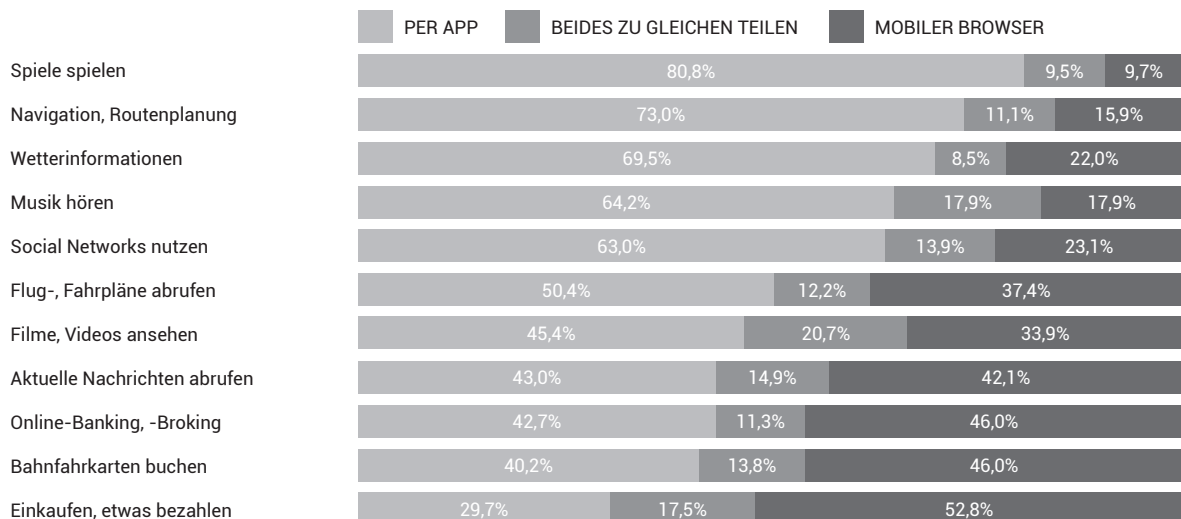


Tabelle 4: App vs. mobiler Browser, Quelle: APPRUPT Studie *App-Nutzung und Kriterien für den App-Download*, Seite 2

Browser-Spiele sind beispielsweise meist sehr ressourcenhungrig und nicht für mobile Endgeräte optimiert, jedoch lässt sich sehr gut etwas bei *Amazon* über den mobilen Browser bezahlen, sodass eine zusätzliche App redundant wird. Die Gleichverteilung bei Nachrichten zeigt deutlich, dass beide Lösungen gleichermaßen effektiv sind. Hier entscheiden persönliche Vorlieben.

Webseiten verzeichnen einen steten Anstieg an Seitenzugriffen via mobilen Browser. Der Anteil ist dabei stark von der jeweiligen Zielgruppe abhängig. Laut einer Studie der Firma *Webtrekk* soll in Deutschland jedoch bereits jeder zehnte Zugriff auf die von dem Unternehmen analysierten Seiten von einem mobilen Browser aus geschehen.<sup>32</sup> Bei der bekannten Online-Schule für Webentwicklung *w3schools.com* wurden im August 2012 1,8 Prozent der Zugriffe von mobilen Browser getätigt, ein Wert, der sich

31 Vgl. APPRUPT GmbH: *App-Nutzung und Kriterien für den App-Download*, Seite 2

32 Vgl. Webtrekk: *Webtrekk Deutsche Webstatistik 3. Quartal 2012*, 02.10.2012, [http://www.webtrekk.com/fileadmin/pdf/pm/PM\\_2012/Webtrekk\\_Studie\\_2012-Q3\\_DE.pdf](http://www.webtrekk.com/fileadmin/pdf/pm/PM_2012/Webtrekk_Studie_2012-Q3_DE.pdf) (29.10.2012)

seit Beginn der Erhebung anfang 2011 monatlich immer um ungefähr 0,1 Prozent erhöht hat.<sup>33</sup> Diese beiden Beispiele zeigen, dass die Nutzer mobilen Browser mittlerweile eine noch kleine, aber wachsende Gruppierung darstellt, die das Internet benutzt.

## 2.4 Unterscheidungsmerkmale der Endgeräte in Hinblick auf plattformübergreifene Webentwicklung

Nachdem nun bewiesen wurde, dass die Notwendigkeit besteht, Smartphones und Tablets bei der Webentwicklung nicht mehr zu ignorieren, stellt sich die Frage, welche Konsequenzen das hat. Doch um optimierte Webseiten für jede dieser Plattformen entwickeln zu können, muss man sich der technischen Unterschiede bewusst werden. Nachfolgend sollen daher die wichtigsten Aspekte zusammengetragen werden.

Der wohl entscheidenste Faktor, der eine Verminderung der Benutzerfreundlichkeit gewöhnlicher Webseiten auf den jeweiligen Endgeräten zur Folge hat, sind die Unterschiede in der Bildschirmgröße. Eine Webseite, die für große PC- und Laptopbildschirme entwickelt wurde, kann nicht ebenso gut auf kleinen Tablet- und Smartphonebildschirmen funktionieren.

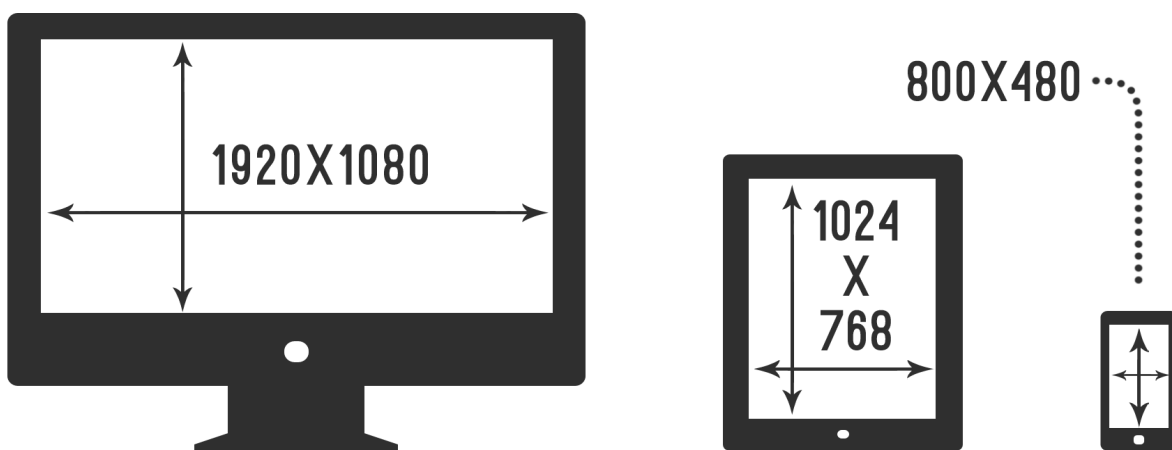


Abb. 9: Gegenüberstellung der verschiedenen Bildschirmauflösungen

33 Vgl. w3schools: OS Platform Statistics,  
[http://www.w3schools.com/browsers/browsers\\_os.asp](http://www.w3schools.com/browsers/browsers_os.asp) (29.10.2012)

Die meisten größeren Tablets haben eine Auflösung von 1280 x 800 Pixeln. Dies entspricht auch den Abmessungen vieler Laptops und stellt für die Webentwicklung somit noch kein Problem dar. Jedoch sind kleinere Geräte mit Auflösungen wie 1024 x 768 Pixeln genauso verbreitet - was in der PC-Welt einer Monitorgröße entspricht, die kaum mehr von Bedeutung ist.<sup>34</sup> Smartphones, die ebenfalls zum Surfen im Internet verwendet werden, haben heute überwiegend eine Auflösung von 800 x 480 Pixeln (z.B.: *Samsung Galaxy S II*). Jedoch besitzen ältere und bestimmte Miniatur-Modelle (z.B.: *Samsung Galaxy Mini*) noch kleinere Abmessungen.<sup>35</sup> Auf diesen Geräten kann eine für große Bildschirme optimierte Webseite nicht den selben Grad an Benutzerfreundlichkeit bieten.

Im Zusammenhang mit den Pixelabmessungen gibt es bei den Bildschirmen von Tablet von Smartphone eine zweite Entwicklung, die für Webgestaltung eine große Rolle spielt. Die Pixeldichte der Bildschirme ist signifikant höher als die der Computer- und Laptopmonitore. Bewegte sie sich bei letzteren noch in Bereichen zwischen 70 und 100 ppi,<sup>36</sup> so sind bei Smartphones und Tablets Pixeldichten von über 200 ppi keine Seltenheit. So hat das *Samsung Galaxy SII* eine Dichte von 217ppi, das neue *iPad 4* von *Apple* mit seinem Retina-Display sogar 264 ppi. Hier befindet sich eine Auflösung von 2048 x 1536 Pixeln auf einem gewöhnlichen 9,7 Zoll Bildschirm. In Zukunft sind bereits Bildschirme mit über 400ppi geplant.<sup>37</sup> Die Folge ist, dass die Webseite auf diesen Bildschirmen kleiner abgebildet wird, Grafiken können unkenntlich werden, Schrift unleserlich.

Ein weiterer Faktor sind die Unterschiede in der Bedienung. Werden Webseiten für gewöhnlich mittels Maus und Tastatur gesteuert, kommt bei Smartphones und Tablets eine Steuerung per Hand über ein Touchscreen zum Einsatz. Dieser Aspekt spielt

---

34 Vgl. w3schools: Browser Display Statistics,  
[http://www.w3schools.com/browsers/browsers\\_display.asp](http://www.w3schools.com/browsers/browsers_display.asp) (29.10.2012)

35 Vgl. binvisions: List of Tablet and Smartphone Resolutions and Screen Sizes,  
<http://www.binvisions.com/articles/tablet-smartphone-resolutions-screen-size-list/> (29.10.2012)

36 Vgl. Prismo: Desktop LCD Display Comparison,  
<http://www.prismo.ch/comparisons/desktop.php> (29.10.2012)

37 Vgl. androidnext.de: LG: Über-Retina-Display mit 1080p auf 5 Zoll und 440 ppi gezeigt, 28.05.2012,  
<http://www.androidnext.de/news/lg-display-1080p-5-zoll-440-ppi/> (29.10.2012)



für eine optimierte Webseite vor allem bei Detailfragen eine große Rolle. So fällt beispielsweise bei der Handsteuerung der Rollover-Effekt - wenn der Mauszeiger über einem Link positioniert wird - weg, da es hier keinen Mauszeiger mehr gibt und Buttons direkt mit dem Finger angewählt werden. Im Hinblick auf eine optimierte Bedienung lässt sich die Bedienung bestimmter Webelemente wie Bilder-Slideshows auf Touchscreens durch Wischen (Abbildung 10) besser bedienen als über das Drücken bestimmter Buttons.

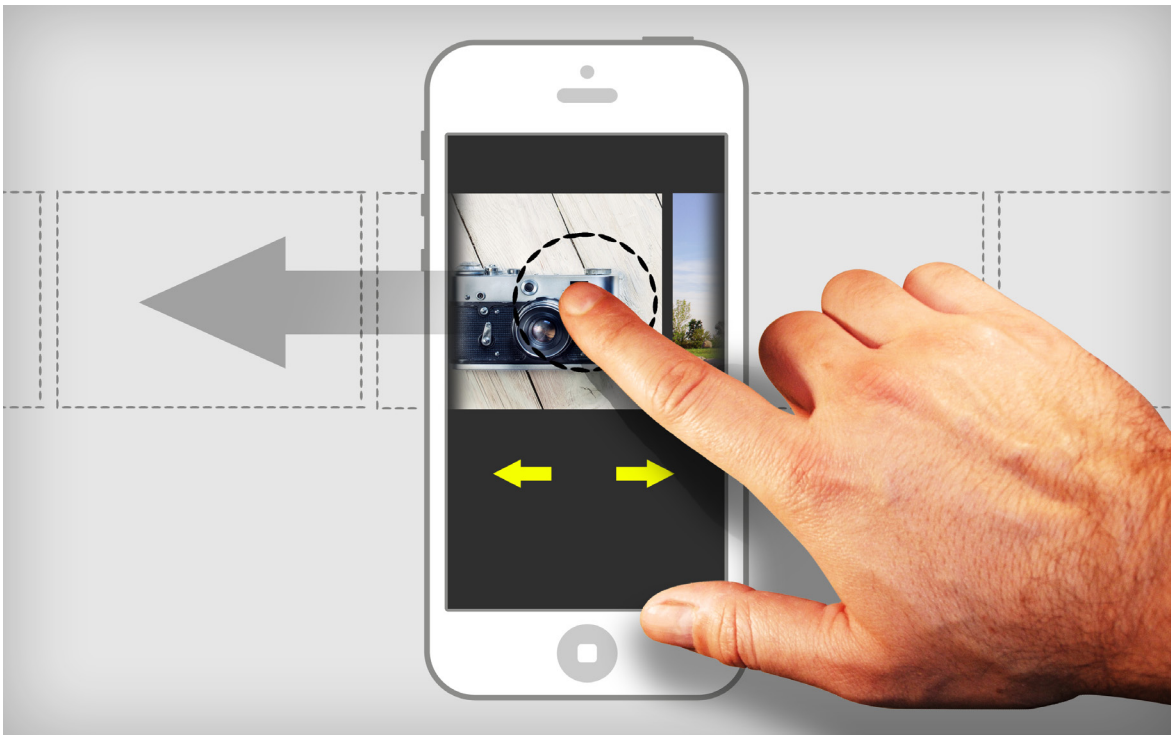


Abb. 10: Wischtechnik auf Touchscreens, angewandt an einer Web-Slideshow

Ein letzter technischer Aspekt, auf den hier eingegangen werden soll, sind die Unterschiede in der Internetverbindung. Tablets und Smartphones sind mobile Geräte und so lang sie nicht auf ein WLAN zugreifen können, nutzen diese je nach Empfang und technischen Möglichkeiten wie *EDGE* oder *HSPA+*, um auf das Internet zugreifen zu können. Die Technologien sind entweder langsam oder preislich an begrenzte Flatrates gebunden. Bei der Gestaltung einer Webseite sollte man diesen Aspekt nicht ignorieren, wenn schnelle Ladezeiten erreichen werden sollen.



## 3 Aktuelle WebStandards

### 3.1 HTML5

HTML5 gilt als zukünftiger Standard für die Entwicklung moderner Internetseiten. Es bietet vielfältige neue Funktionalitäten wie Video- und Audiofunktionen, die Verwendung lokalen Speichers oder dynamische 2D- und 3D-Grafiken. Viele der Features, die nun ausschließlich mittels HTML5 realisiert werden können, ließen sich bisher nur durch den Einsatz von JavaScript oder Flash umsetzen.

Leider können viele ältere Browser HTML5 nicht verstehen. Trotzdem nutzen immer mehr Webseiten den neuen Standard, da durch verschiedene Fallback-Strategien ein ausreichend großes Fangnetz eingerichtet werden kann (Vgl. Punkt 3.3 *Neue Neue Einsatzmöglichkeiten von Javascript*). So sind Nutzer älterer Browser nicht vom Surferlebnis ausgeschlossen. Der Suchmaschine builtwith.com sind mittlerweile über 6,8 Millionen Webseiten bekannt, die HTML5 verwenden - mit steigender Tendenz.<sup>38</sup>

Gearbeitet wird an HTML5 bereits seit 2004 mit dem Ziel, dass vor allem die Nutzer der Sprache - z. B. Browser- oder Webentwickler - an dessen Entwicklung beteiligt sind.<sup>39</sup> Die Entwicklung ist bis heute nicht abgeschlossen - die Technologie wird aber, wie bereits beschrieben, immer häufiger eingesetzt. Einer der ersten großen Unterstützer des Webstandards war die Firma *Apple*. Dessen mobiles Betriebssystem *iOS* setzte den Fokus auf HTML5 und unterstütze den Einsatz von Flash auf Webseiten nicht mehr.

Heute kann HTML5 als etablierter Webstandard betrachtet werden. Die wichtigsten neuen Funktionen sollen daher in den folgenden Kapiteln kurz vorgestellt werden.

---

38 Vgl. builtwith.com: HTML5 DocType Usage Statistics,  
<http://trends.builtwith.com/docinfo/HTML5-DocType> (29.10.2012)

39 Vgl. Lee, Tim Berners: Reinventing HTML, 27.06.2006,  
<http://dig.csail.mit.edu/breadcrumbs/node/166> (29.10.2012)

### 3.1.1 Dokumententyp und Semantik

Damit eine Seite ordnungsgemäß verarbeitet werden kann, benötigt sie einen Doctype. Dieser sagt dem Browser, um was für eine Art Dokument es sich handelt und wie er dieses darzustellen hat. Im Vergleich zu älteren HTML-Versionen ist der Doctype eines HTML5-Dokumentes sehr einfach anzugeben:

```
<!doctype html>
```

Der Doctype eines HTML5-Dokuments wird von jedem Browser, auch älteren Internet Explorer-Versionen, verstanden.

Auch die semantische Struktur wurde abgespeckt und an die Bedürfnisse der Webentwickler angepasst. Gleichzeitig ist HTML5 in seiner Syntax lockerer geworden und verzeiht, anders als beispielsweise das sehr strenge XHTML, Unsauberkeiten wie wechselnde Groß- und Kleinschreibung oder nicht geschlossene HTML-Elemente.<sup>40</sup>

Ein HTML5-Dokument lässt sich neben dem Doctype sehr leicht auch am strukturellen Aufbau erkennen. Verantwortlich dafür sind die neuen semantischen Elemente. Wurden ursprünglich die einzelnen Sektionen einer Webseite wie der Header, der Footer, oder die Navigation alle in einem `<div>`-Container mit entsprechender ID untergebracht, sind dafür nun die Elemente `<header>`, `<nav>` und `<footer>` vorgesehen.<sup>41</sup> Diese drei Elemente zeigen, dass eines der Ziele von HTML5 ist, die Seite mit einer klareren Struktur zu versehen. Eine Navigation oder ein Header sind universelle Bereiche, die auf jeder Webseite vorhanden sind, jedoch nicht unmittelbar zum eigentlichen Inhalt gehören. Mit Hilfe der neuen Elemente können diese nun in ihrer Funktion eindeutig definiert werden, so dass es beispielsweise Suchmaschinen leichter fällt, diese zuzuordnen.

---

40 Vgl. Kröner, Peter: HTML 5, Seite 57

41 Vgl. W3 Consortium: HTML: The Markup Language (an HTML language reference), <http://www.w3.org/TR/html-markup/elements.html> (29.10.2012)

Ein weiteres neues Element ist `<section>`. Seine Aufgabe ist es, eine Webseite in verschiedene Sinn-Abschnitte zu unterteilen.<sup>42</sup> Das können einzelne Kapitel eines Textes sein, aber auch ganze Themenbereiche der Webseite. Das `<section>`-Element kann mit einführenden Inhalten in Form des bereits erwähnten `<header>`-Elements eingeleitet und mit abschließenden Inhalten durch ein `<footer>`-Element beendet werden. Hier werden die sinnbezogene Verwendungszwecke der neuen Elemente deutlich. Ein Beispiel für eine Nachricht, welches die neuen Elemente sinngemäß nutzt, könnte wie folgt aussehen:<sup>43</sup>

```
<section>
  <header>
    <h1>Ueberschrift</h1>
    <p>Datum</p>
  </header>
  <p>Inhalt der Nachricht</p>
  <footer>
    <p>Kategorie der Nachricht</p>
    <p>Anzahl der Kommentare</p>
  </footer>
</section>
```

Es gibt einige weitere neue HTML5-Elemente. Mit `<article>` werden abgeschlossene Inhalte ausgezeichnet, im obigen Beispiel hätte somit auch an Stelle des `<section>`-Elements ein `<article>` zum Einsatz kommen können.<sup>44</sup> `<aside>` repräsentiert Inhalte, die eine Ergänzung für ein bestimmtes Element darstellen.<sup>45</sup> Mit `<hgroup>` lassen sich mehrteilige Überschriften zusammenfassen.<sup>46</sup> Diese sollen als Beispiele genügen.

---

42 Vgl. W3 Consortium: HTML: The Markup Language (an HTML language reference) - section, <http://www.w3.org/TR/html-markup/section.html#section> (29.10.2012)

43 Vgl. Kröner, Peter: HTML 5, Seite 116

44 Vgl. W3 Consortium: HTML: The Markup Language (an HTML language reference) - article, <http://www.w3.org/TR/html-markup/article.html#article> (29.10.2012)

45 Vgl. W3 Consortium: HTML: The Markup Language (an HTML language reference) - aside, <http://www.w3.org/TR/html-markup/aside.html#aside> (29.10.2012)

46 Vgl. W3 Consortium: HTML: The Markup Language (an HTML language reference) - hgroup, <http://www.w3.org/TR/html-markup/hgroup.html#hgroup> (29.10.2012)

### 3.1.2 Das <canvas>-Element

Das <canvas>-Element ist eines der populärsten Features von HTML5. Canvas bedeutet zu Deutsch Leinwand und ist ein grafisches Element mit enormem Potential. Es kann zur Implementierung von 2D- und 3D-Grafiken, Animationen oder ganzer interaktiver Anwendungen genutzt werden und arbeitet eng mit JavaScript zusammen. Das Element wird mittlerweile von vielen Browsern unterstützt (Tabelle 5) und es existieren verschiedene Fallback-Strategien, die das gewünschte Ergebnis auch auf älteren Browsern nachbilden lässt. Das volle Möglichkeitenspektrum des <canvas>-Elementes würde den Rahmen dieser Arbeit sprengen. Daher soll hier nur kurz auf die grundlegende Handhabung eingegangen werden.

IE	FIREFOX	CHROME	SAFARI	OPERA	IOS SAFARI	OPERA MINI	ANDROID	BLACKBERRY
9.0	14.0	21.0	5.1	12.0	3.2	5.0-7.0	2.1	7.0

Tabelle 5: Browsersupport <canvas>-Element, ab welcher Version - Quelle: <http://caniuse.com>

Anfangs wird immer eine Leinwand definiert. Dieses <canvas>-Element ist zu Beginn leer, besitzt also weder Rahmen noch Inhalte. Praktischerweise sollte das Element daher mit einer Höhe und Breite sowie einer entsprechenden ID versehen werden, um es via JavaScript ansprechen zu können.

```
<canvas width="400" height="300" id="beispiel-canvas" ></canvas>
```

Was nun auf dieser Bühne geschehen soll, wird vollkommen über JavaScript gesteuert. Ein einfaches Rechteck einzufügen ließe sich beispielsweise durch folgendes Script bewerkstelligen:

```
<script>
    function drawCanvas() {
        var canvas = document.getElementById('beispiel-canvas');
        if(canvas.getContext) {
            var context = canvas.getContext('2d');
```

```
        context.fillStyle = 'rgb(255, 0, 0)';  
        context.fillRect(50, 50, 50, 50);  
    }  
}  
</script>  
<body onload="drawCanvas()">  
    <canvas width="400" height="300" id="beispiel-canvas" ></canvas>  
</body>
```

Es wird eine Funktion `drawCanvas()` definiert, die nach dem Laden der Seite gestartet wird. Dieser wird als erstes das `<canvas>`-Element durch die entsprechende ID zugewiesen. Anschließend wird die Variable `canvas` über die Zuweisung `var context = canvas.getContext('2d');` als 2D Werkzeug initialisiert. Es folgt Füllfarbe und Abmessung des Rechtecks, wobei die vier Werte jeweils den Abstand des Rechtecks zum Rand darstellen.<sup>47</sup>

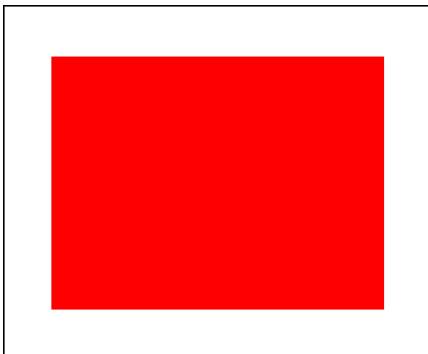


Abb. 11: Rechteck auf einem `<canvas>`-Element, jeweils 50 Pixel Abstand zum Rand

In ein `<canvas>`-Element lassen sich jedoch weit komplexere Inhalte einfügen. So werden komplette HTML5-Spiele, welche beispielsweise auf Seiten wie `html5games.com` oder `html5spiele.de` angeboten werden, innerhalb eines `<canvas>`-Elementes über JavaScript gesteuert. Selbst der 3D-Shooter *Quake 2* wurde bereits erfolgreich nach HTML5 konvertiert.<sup>48</sup>

---

<sup>47</sup> Vgl. Kröner, Peter: HTML 5, Seite 358

<sup>48</sup> Vgl. Haustein, Stefan u. a.; Quake II GWT Port, <http://code.google.com/p/quake2-gwt-port/> (29.10.2012)

### 3.1.3 Multimedia-Elemente

Multimediale Inhalte wie Videos oder Musik sind im Internet äußerst präsent. Webseiten wie *Youtube*, *Grooveshark* oder *Soundcloud* sind sogar ausschließlich auf die Bereitstellung dieser Elemente ausgelegt. Bisher gab es jedoch keinen Standard, der das Einbinden solcher Inhalte möglich gemacht hat. Man war auf externe Technologien wie Flash angewiesen. Diese stellte jedoch schon immer eine suboptimale Lösung dar, da der Nutzer auf seinem Computer das Flash-Plugin installiert haben musste und dieses enorme Ressourcen beanspruchte. Die Nachfrage nach einer Alternative war also groß. Mit HTML5 wurde diese geliefert.

Es definiert dafür zwei neue Elemente. Mittels `<audio>` und `<video>` kann auf HTML5-fähigen Browsern nun Ton und Film unkompliziert eingebunden werden.<sup>49 50</sup> Für Videos sind drei Formate erlaubt: MP4, WebM oder Ogg. Eingebunden wird ein Video dabei beispielsweise wie folgt:

```
<video width="400" height="300" controls="controls">
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
  <p>Ihr Browser unterstützt keine HTML5 Video Elemente.</p>
</video>
```

Dieses Beispiel hat verschiedene Parameter, dessen Funktionen kurz erläutert werden sollen. Innerhalb des `<video>`-Elementes hat man die Möglichkeit, die Abmessungen des Players mittels `width` und `height` zu bestimmen. Durch den Parameter `controls` werden verschiedene Steuerungselemente wie Start- und Stop-Buttons in den Player eingefügt. Es folgt die Angabe der Quelle des Videos, wobei hier mehrere Formate für die unterschiedlichen Browser ratsam sind. Gleichzeitig sollte innerhalb des Elements ein entsprechender Hinweis platziert werden, falls Video-Elemente vom Browser nicht

---

49 Vgl. W3 Consortium: `<audio>` Element,  
<http://www.w3.org/wiki/HTML/Elements/audio> (29.10.2012)

50 Vgl. W3 Consortium: `<video>`-Element,  
<http://www.w3.org/wiki/HTML/Elements/video> (29.10.2012)

unterstützt werden. Möglich ist auch die Verwendung des `poster`-Attributs, über das ein Bild im Fenster des Players angezeigt werden kann, wenn dieser noch nicht geladen oder gestartet wurde.

Das Einbinden einer Audio-Datei funktioniert auf die gleiche Weise, wobei an Stelle des `<video>`-Elementes ein `<audio>`-Element zum Einsatz kommt. Unterstützte Formate sind MP3, Wav oder Ogg.

### 3.1.4 Das SVG-Format

SVG ist ein Grafikformat und steht für *Scalable Vector Graphics*. Es handelt sich also um ein Format für Vektorgrafiken, die sich, im Gegensatz Pixelgrafiken, durch ihre verlustfreie Skalierbarkeit auszeichnen. SVG nutzt die Auszeichnungssprache XML und definiert mit deren Hilfe Pfade, Formen, Schriftarten und Farben der Grafik. HTML5 kann dieses Format nun ebenfalls korrekt interpretieren und nutzen.

Für plattformübergreifende Webentwicklung ist der Einsatz auflösungsunabhängiger Grafiken sehr nützlich. Das Heranzoomen von Webseiten ist gängige Praxis vor allem auf mobilen Geräten. Stark vergrößerte Pixelgrafiken werden dann jedoch unscharf (Abbildung 12).

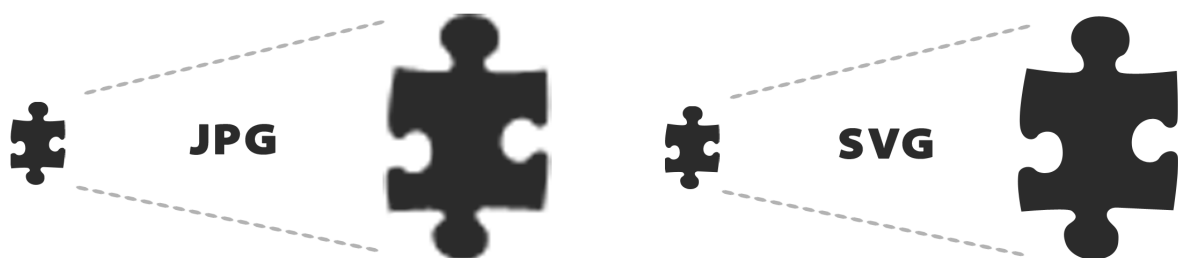


Abb 12:: Skalierungsunterschiede zwischen JPG- und SVG- Format

Komplexe SVG-Grafiken werden üblicherweise mittels eines Grafikprogrammes wie Adobe Illustrator erstellt. Es ist jedoch auch möglich, einfache Formen direkt im HTML5-Markup zu definieren. Innerhalb eines `<svg>`-Elementes lassen sich so Kreise, Rechtecke und mehr unkompliziert realisieren:

```
<svg id="beispiel-svg-1">
  <rect width="200" height="200" fill="red" />
</svg>
```

Im obigen Beispiel wird so ein 200x200 Pixel großer Kreis erstellt. Der Einsatz von des SVG-Formates wird von allen modernen Browsern unterstützt. Für ältere gibt es außerdem ausreichende Fallback-Mechanismen, die den Einsatz der Vektorgrafiken sicher machen.<sup>51</sup>

IE	FIREFOX	CHROME	SAFARI	OPERA	IOS SAFARI	OPERA MINI	ANDROID	BLACKBERRY
9.0	14.0	21.0	5.1	12.0	3.2	5.0-7.0	3.0	7.0

Tabelle 6: Ab welcher Version wird SVG-Format von Browsern unterstützt - Quelle: <http://caniuse.com>

### 3.1.5 Weitere Features von HTML5

HTML5 unterstützt außerdem verschiedene JavaScript-APIs, die die Verwendung weiterer interessanter Funktionen ermöglichen. So lassen sich mittels der Geolocation-API zuverlässige Informationen über den Standort des Nutzers ermitteln und für verschiedene Zwecke wie ortsbezogene Werbebanner nutzen.<sup>52</sup> Mit der Drag&Drop-Api lassen sich nun auch Drag&Drop-Funktionen ohne den Einsatz von JavaScript-Bibliotheken wie *jQuery* oder *Mootools* umsetzen. Gemeint ist damit die Möglichkeit, Elemente per Mausklick auf dem Browser hin- und herzuschieben (Abbildung 13).

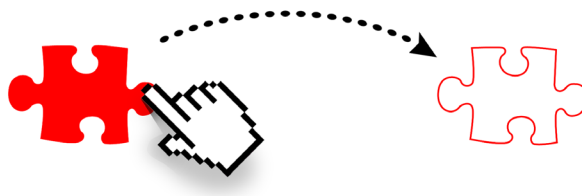


Abb 13: Veranschaulichung der Drag&Drop-Funktionsweise

51 Vgl. W3 Consortium: Inline SVG in HTML5 and XHTML, <http://dev.w3.org/SVG/proposals/svg-html/svg-html-proposal.html> (29.10.2012)

52 Vgl. Kröner, Peter: HTML 5, Seite 217



Ein weiteres interessantes Feature von HTML5 ist der Local Storage, dessen Verwendung wieder den Einsatz von JavaScript erfordert. Mit Hilfe des Local Storage lassen sich bis zu 5MB an Daten pro Domain auf dem lokalen Rechner speichern.<sup>53</sup> Jeder Browser hat hier seinen eigenen Speicherplatz. Nutzen kann man den Local Storage z.B. um Formulardaten zu speichern. Wenn der Anwender den Zurück-Button benutzt oder eine Aktualisierung der Webseite durchführt, sind die von ihm getätigten Eingaben nicht weg, sondern können damit wieder angezeigt werden. Das senkt das Frustrationspotenzial beim Kunden und ist dazu noch sehr einfach einzubinden.

Zuletzt sollen die neuen HTML5-Inputelemente Erwähnung finden, die die Erstellung komplexerer Formulare ermöglichen. Zu den neuen Eingabetypen gehören beispielsweise `number`, `tel`, `range` oder `search`. Mit Hilfe dieser neuen Typen kann der Entwickler bereits auf Clientseite Eingabefelder auf bestimmte Eingabewerte überprüfen, ohne aufwendige JavaScript Funktionen zu verwenden. Die Attribute `placeholder` und `required` werden dabei mittlerweile von fast allen Browsern unterstützt. Ersteres erlaubt es Input-Feldern einen zusätzlichen Text z.B. zur besseren Verständlichkeit der geforderten Eingabe zuzuweisen. Der Text wird angezeigt, wenn sich kein Text im Textfeld befindet und wird direkt entfernt, sobald ein Text eingegeben oder das Feld fokussiert wird. Das `required`-Attribut hingegen überprüft bei Input-Feldern, ob eine Eingabe gemacht wurde bevor das Formular an den Server übertragen wird. Wenn eine Eingabe für ein Feld mit dem Attribut `required` vergessen wurde, wird eine Fehlermeldung angezeigt.<sup>54</sup>

---

53 Vgl. W3 Consortium: Web Storage,  
<http://www.w3.org/TR/webstorage/> (29.10.2012)

54 Vgl. W3 Consortium: Input Types,  
<http://www.w3.org/TR/html-markup/input.number.html> (29.10.2012)

## 3.2 CSS

CSS steht für *Cascading Style Sheets*. Es ist eine Stil-Sprache, die in Verbindung mit HTML oder XML zum Einsatz kommt. Hat HTML die Aufgaben, Inhalte zu strukturieren, so ist es mittels CSS möglich, diese Struktur in Form, Positionierung und Gestaltung anzupassen. Einem HTML-Element werden dazu verschiedene CSS-Regeln zugewiesen, die beispielsweise den in dem Element befindlichen Text bezüglich Größe, Farbe und Schriftart formatieren. Gleichzeitig kann das gesamte Layout einer Webseite gestaltet und gesteuert werden. Somit macht CSS eine plattformübergreifende und -unabhängige Webgestaltung überhaupt erst möglich.

CSS wurde 1994 erstmals von Håkon Wium Lie vorgeschlagen.<sup>55</sup> Die 1996 veröffentlichte *CSS Level 1 Recommendation* enthielt jedoch vorerst nur grundlegende gestalterische Möglichkeiten.<sup>56</sup> Mit den steigenden Anforderungen an das Web und immer komplizierteren Webseiten, stieß CSS1 schnell an seine Grenzen. So war es beispielsweise nicht möglich, separate Gestaltungsregeln für die jeweiligen Betrachtungsmedien wie Bildschirm oder Print festzulegen. Das 1998 veröffentlichte *CSS Level 2* enthielt daher viele neue Features wie das Festlegen von entsprechenden Medientypen.<sup>57</sup> CSS2 sowie CSS2.1 wird heute von modernen Browsern fast vollständig verstanden und korrekt dargestellt.

Seit dem Jahr 2000 wird bereits an CSS3 gearbeitet, jedoch hat die Entwicklung und Verbreitung moderner Browser erst in den letzten Jahren einen Stand erreicht, die dessen Einsatz möglich macht. CSS3 bietet dabei einfache Lösungen für Gestaltungselemente, die ursprünglich nur sehr aufwändig umgesetzt werden konnten. Und selbst Animationen lassen sich mittlerweile komplett ohne JavaScript ausschließlich mit CSS3 umsetzen. Dieses und weitere Features sollen in den folgenden Kapiteln genauer vorgestellt werden.

---

55 Vgl. Lie, Håkon Wium: Cascading Style Sheets, 2005, <http://people.opera.com/howcome/2006/phd/#ch-introduction> (29.10.2012)

56 Vgl. W3 Consortium: Cascading Style Sheets, level 1, 17 Dec 1996, <http://www.w3.org/TR/REC-CSS1/> (29.10.2012)

57 Vgl. W3 Consortium: Media Types, <http://www.w3.org/TR/2002/WD-CSS21-20020802/media.html> (29.10.2012)

### 3.2.1 Rahmen

CSS3 bietet für die Gestaltung des Rahmens eines HTML-Elementes einige neue Möglichkeiten, die sich früher nur sehr aufwändig und umständlich umsetzen ließen. Eines der am häufigsten eingesetzten Gestaltungsmittel auf einer Webseite ist der Einsatz abgerundeter Ecken. Es ist gleichzeitig eines der besten Beispiele um aufzuzeigen, wie CSS3 den Arbeitsaufwand bei der Umsetzung einer Webseite reduziert. Ließen sich abgerundete Ecken früher nur über das Einbinden verschiedener, in einem Bildbearbeitungsprogramm erstellter JPG-Grafiken umsetzen<sup>58</sup>, lässt sich das selbe Resultat nun schnell und unkompliziert mit der CSS-Regel `border-radius` erzielen. Selbst große Portale wie Youtube oder Twitter machen von der Regel in ihrem Webdesign gebrauch (Abbildung 14).

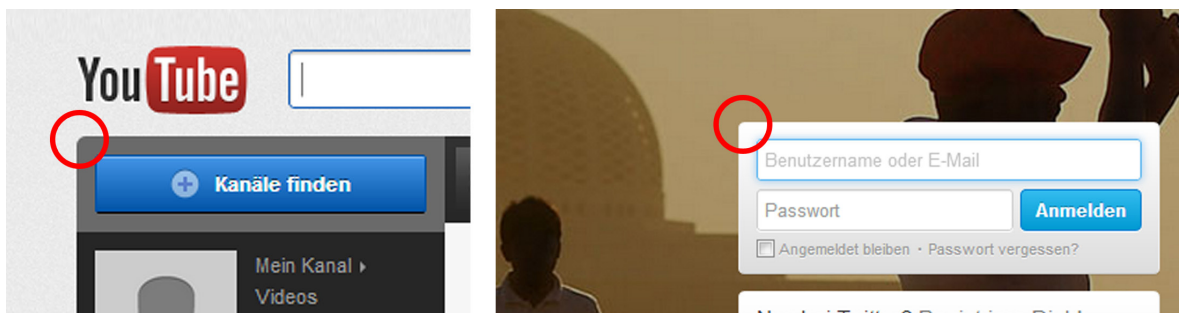


Abb 14: Einsatz abgerundeter Ecken auf Youtube und Twitter

Mittels `border-radius` lässt sich ein Radius für alle vier Ecken eines HTML-Elementes definieren.<sup>59</sup> Dieser wird mittels eines entsprechenden Parameters angegeben. Dabei kann es sich beispielsweise um einen Pixel- oder Prozentwert handeln. Die korrekte Schreibweise lautet wie folgt:

```
#beispiel-1 { border-radius: 15px; }
```

58 Vgl. Willison, Simon: Rounded Corners with CSS and JavaScript, 28.05.2004, <http://www.sitepoint.com/rounded-corners-css-javascript/> (29.10.2012)

59 Vgl. W3 Consortium: CSS Backgrounds and Borders Module Level 3, <http://www.w3.org/TR/css3-background/#the-border-radius> (29.10.2012)

Das HTML-Element `#beispiel-1` erhält durch jene Regel einen kreisförmigen Eckradius von `15px` in jeder Ecke (Abbildung 15). Es gibt jedoch auch die Möglichkeit, jeder Ecke einen separaten Eckradius zuzuweisen. Die dazugehörigen CSS-Regeln lauten wie folgt:

```
#beispiel-2 {  
    border-top-left-radius:    5px;  
    border-top-right-radius:   10px;  
    border-bottom-right-radius: 15px;  
    border-bottom-left-radius: 20px;  
}
```

Die Kurzform lautet:

```
#beispiel-2 { border-radius: 5px 10px 15px 20px; }
```

Der CSS-Regel `border-radius` folgen in der Kurzschreibweise vier Parameter. Begonnen wird dabei mit der linken oberen Ecke. Von dieser Ecke ausgehend wird im Uhrzeigersinn jeder weiteren Ecke ihr separater Radius zugewiesen. Als Resultat erhält das HTML-Element `#beispiel-2` vier abgerundete Ecken unterschiedlicher Ausprägung (Abbildung 15). Zusätzlich hat man die Möglichkeit, jeder Ecke zwei verschiedene Parameter zuzuweisen, um eine ovale Eckenform zu erhalten. Die korrekte Schreibweise lautet:

```
#beispiel-3 {  
    border-top-left-radius:    5px 15px;  
    border-top-right-radius:   10px 20px;  
    border-bottom-right-radius: 15px 5px;  
    border-bottom-left-radius: 20px 10px;  
}
```

Die Kurzform lautet:

```
#beispiel-3 { border-radius: 5px 10px 15px 20px / 15px 20px 5px 10px; }
```

Die Reihenfolge der Parameter richtig sich wiederrum nach dem Uhrzeigersinn beginnend mit der linken oberen Ecke (Abbildung 15).



Abb 15: Veranschaulichung der vorangehenden Beispiele #beispiel-1, #beispiel-2 und #beispiel-3

Tabelle 8 zeigt, dass die CSS-Regel `border-radius` mittlerweile von fast allen Browsern, sei es auf PC oder mobilen Geräten, unterstützt und korrekt dargestellt wird. Dieses und auch einige der nachfolgend vorgestellten CSS3-Features werden nicht von älteren Versionen des Internet Explorers verstanden, jedoch gibt es verschiedene JavaScript-Lösungen, die hier Abhilfe schaffen (Vgl. Punkt 3.3 *Neue Einsatzmöglichkeiten von JavaScript*).

IE	FIREFOX	CHROME	SAFARI	OPERA	IOS SAFARI	OPERA MINI	ANDROID	BLACKBERRY
9.0	14.0	21.0	5.1	12.0	3.2	-	2.1	7.0

Tabelle 7: Ab welcher Version wird `border-radius` von Browsern unterstützt - Quelle: <http://caniuse.com>

Eine weitere Gestaltungsmöglichkeit für Rahmen bietet sich mit der CSS-Eigenschaft `border-image`.<sup>59</sup> Diese erlaubt es, anstelle der gewöhnlichen Rahmen-Stile wie `solid` oder `dotted` ein individuell gestaltetes Bild als Rahmenelement zu definieren. Mit dieser Eigenschaft lassen sich bestimmte Layouts leicht umsetzen, welche vorher nur umständlich über die CSS-Eigenschaft `background-image` realisiert werden konnten.

59 Vgl. W3 Consortium: CSS Backgrounds and Borders Module Level 3, <http://www.w3.org/TR/css3-background/#border-images> (29.10.2012)

Der CSS-Regel werden dabei drei Parameter zugewiesen:

```
border-image-source:    url(beispiel-bild.png);  
border-image-slice:    30px;  
border-image-repeat:    repeat;
```

Die Kurzform lautet:

```
border-image: url(beispiel-bild.png) 30px repeat;
```

Zuerst wird festgelegt, welches Bild als Quelle für den Rahmen genommen werden soll. Außerdem muss festgelegt werden, welche Bereiche des Bildes an welchen Positionen platziert und wiederholt werden sollen. Dies geschieht zum einen über die Gestaltung des Bildes. Zum anderen wird über CSS-Parameter festgelegt, wie es verwendet werden soll. Das Bild wird dabei in acht Bereiche gegliedert, die jeweils die grafischen Elemente für die vier Ecken sowie die vier Seiten enthalten.

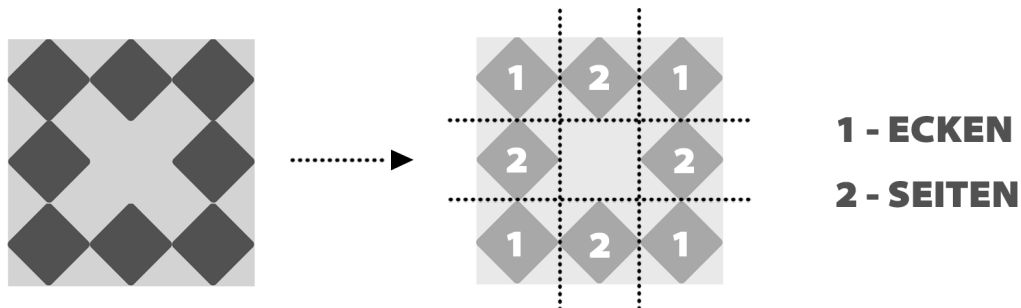


Abb 16: Beispiel für *border-image* - Quelle: w3.org

Abbildung 16 zeigt, wo sich die einzelnen Rahmenbereiche befinden und wie die Grafik gesliced werden müsste. Dies wird mit dem zweiten Parameter bewerkstelligt. Da in unserem Beispielbild eine gleichmäßige Teilung vorliegt, reicht in diesem Fall die Angabe eines einzelnen Parameters (30px). Erfolgt die Teilung ungleichmäßig, benötigt man wiederum vier Parameter, die die jeweiligen Teilungsabstände vom Bildrand aus gesehen definieren. Begonnen wird dabei mit der oberen Kante. Die folgenden Kanten definieren sich entsprechend im Uhrzeigersinn.

Der letzte Parameter beschreibt das Verhalten der Grafik an den Seiten. Mögliche Werte sind `repeat`, `stretch` und `round`. Gibt man `repeat` an, wird die Grafik an den Seiten auf der gesamten Länge wiederholt, `stretch` dehnt die Grafik auf die Länge der Seite und `round` wiederholt das Bild auf der Länge der Seite nur oft oft, wie es als Ganzen hineinpasst, die werden dann auf die Länge der Seite gedehnt.

IE	FIREFOX	CHROME	SAFARI	OPERA	IOS SAFARI	OPERA MINI	ANDROID	BLACKBERRY
-	14.0	21.0	5.1	12.0	3.2	-	2.1	7.0

Tabelle 8: Ab welcher Version wird *border-image* von Browsern unterstützt - Quelle: <http://caniuse.com>

Wie in Tabelle 8 deutlich wird, kann die CSS-Eigenschaft *border-image* von vielen Browsern noch nicht korrekt interpretiert werden. Um auf dieses Feature jedoch trotzdem nicht verzichten zu müssen, kann als Fallback-Methode mit der zusätzlichen Angabe eines gewöhnlichen Rahmens wie

```
border: 5px solid red;
```

eine Absicherung eingerichtet werden. So erscheint ein grafischer Ersatz im Falle nicht unterstützter Browser.

### 3.2.2 Farben und Farbverläufe

CSS3 erweitert die möglichen Parameter zur Definition von Farben um entscheidende Möglichkeiten. So können Farben zwar immer noch via Hexadezimalcode oder im RGB-Format definiert werden, jedoch wurden weitere Möglichkeiten hinzugefügt.<sup>60</sup> So lässt sich beispielsweise die Angabe der Farbe Rot im RGB-Format

```
color: rgb(255,0,0);
```

60 Vgl. W3 Consortium: CSS Color Module Level 3, 07.06.2011, <http://www.w3.org/TR/css3-color/#hsl-color> (29.10.2012)

um den Parameter der Transparenz erweitern:<sup>61</sup>

```
color: rgba(255, 0, 0, 1);
```

Der Wert 1 entspricht dabei einer Transparenz von 100 Prozent, ein Wert von 0.5 würde in diesem Fall ein zu 50 Prozent transparentes Rot bedeuten. Gleichzeitig wurde neben dem RGB-Farbspektrum ein komplett neues Farbsystem integriert. So wird beim HSL-Format (*hue-saturation-lightness* - Färbung, Sättigung, Helligkeit) die Farbe Grün wie folgt bestimmt:

```
color: hsl(120, 100%, 50%);
```

Auch diese Angabe kann um den Wert der Transparenz ergänzt werden:

```
color: hsla(120, 100%, 50%, 1);
```

Wie in Tabelle 9 deutlich wird, kann die Angabe eines Farbwertes im HSL-Format, sowie die eines Transparenz-Wertes von fast allen Browsern korrekt dargestellt werden. Transparente Farbwerte lassen sich daher an verschiedensten Stellen einsetzen, sei es in Farbverläufen oder bei Schlagschatten.

IE	FIREFOX	CHROME	SAFARI	OPERA	IOS SAFARI	OPERA MINI	ANDROID	BLACKBERRY
9.0	14.0	21.0	5.1	12.0	3.2	5.0-7.0	2.1	7.0

Tabelle 9: Ab welcher Version wird das HSL-Format sowie Transparenz von Browsern unterstützt

Quelle: <http://caniuse.com>

Weiterhin ist es nun auch möglich, für einen Hintergrund statt einer einzelnen Farbe einen Farbverlauf zu definieren. Diese können linear oder kreisförmig verlaufen.

<sup>61</sup> Vgl. W3 Consortium: W3 Consortium: CSS Color Module Level 3, 07.06.2011, <http://www.w3.org/TR/css3-color/#rgba-color> (29.10.2012)



Ein Verlauf von Blau zu Gelb kann mit folgender CSS-Regel erzeugt werden.<sup>62</sup> Die prozentualen Angaben definieren dabei die Anfangs- und Endpunkte des Verlaufs. (Siehe Abbildung 17)

```
#beispiel-1 {
    background-image: linear-gradient (top, blue 0%, yellow 100%);
}
#beispiel-2 {
    background-image: linear-gradient (top, blue 0%, yellow 50%);
}
#beispiel-3 {
    background-image: radial-gradient (top, blue 0%, yellow 100%);
}
```



Abb 17: Veranschaulichung Farbverläufe - *#beispiel-1*, *#beispiel-2* und *#beispiel-3*

Der CSS-Parameter `linear-gradient` bzw. `linear-gradient` wird derzeit von kaum einem Browser unterstützt, allerdings haben die meisten längst eigene Befehle integriert, um Farbverläufe darzustellen. Dieser lautet beispielsweise auf WebKit basierten Browsern `-webkit-linear-gradient`, auf Browsern von Mozilla `-moz-linear-gradient`. Als Fallback sollte zusätzlich auch ein Bild mit dem entsprechenden Farbverlauf als Hintergrund eingebunden werden. In den unterstützten Browsern wird dieses Bild dann jedoch nicht geladen und Ladezeiten der Webseite reduzieren sich.

IE	FIREFOX	CHROME	SAFARI	OPERA	IOS SAFARI	OPERA MINI	ANDROID	BLACKBERRY
10.0	14.0	21.0	5.1	12.0	3.2	-	2.1	7.0

Tabelle 10: Ab welcher Version werden Farbverläufe von Browsern unterstützt - Quelle: <http://caniuse.com>

62 Vgl. W3 Consortium: CSS Image Values and Replaced Content Module Level 3, 12.06.2012, <http://dev.w3.org/csswg/css3-images/> (29.10.2012)

### 3.2.3 Schlagschatten

CSS3 enthält verschiedene Parameter, um HTML-Container oder auch Textelemente mit Schatten zu versehen. Dies konnte früher nur über das Einbinden unterschiedlicher, transparenter PNG-Dateien realisiert werden, eine dank CSS3 heute obsolete Technik, die höchstens noch als Fallback für ältere Browser zum Einsatz kommt.

CSS3 bietet dabei eine Vielzahl an Gestaltungsmöglichkeiten für die Art und Form des Schattens.<sup>63</sup> Ein Schlagschatten für das HTML-Element `#beispiel-1` könnte so definiert werden:

```
#beispiel-1 { box-shadow: 3px 3px 5px 6px #000; }
```

Der erste Parameter steht dabei für die horizontale Verschiebung, die der Schatten im Verhältnis zum Container besitzen soll. Der zweite Parameter legt die vertikale Verschiebung fest. Diese ersten beiden Parameter sind Pflichtangaben, für die außerdem auch negative Werte zulässig sind. Der dritte Parameter beschreibt den Unschärferadius des Schattens. Je höher der Wert, desto unschärfer wird die Darstellung. Der vierte Parameter beschreibt die Größe des Schattens im Vergleich zum Container. Zuletzt wird noch die Farbe festgelegt. Es ist auch möglich den Schatten nach innen fallen zu lassen. Dazu setzt man vor alle Parameter zusätzlich den Wert `inset`.

```
#beispiel-2 { box-shadow: inset 0px 0px 5px #000; }
```



Abb 18: Veranschaulichung Schatten - `#beispiel-1` und `#beispiel-2`

63 Vgl. W3 Consortium: CSS Backgrounds and Borders Module Level 3, 24.06.2012, <http://www.w3.org/TR/css3-background/#the-box-shadow> (29.10.2012)

Wie schon bei den Farbverläufen benötigen die verschiedenen Browser auch hier entsprechende Kürzel vor dem Parameter `box-shadow`, um Schlagschatten erzeugen zu können.

Es können jedoch nicht nur HTML-Elemente mit einem Schatten versehen werden, auch für reinen Text ist es mit der entsprechenden CSS-Regel `text-shadow` möglich, einen Schatten zu definieren.<sup>64</sup> Zu beachten ist dabei, dass der Text auch ohne den Schatten gut lesbar bleibt, da Textschatten nicht auf allen Browsern dargestellt werden kann (Tabelle 11). Dabei werden `text-shadow` wie auch beim `box-shadow` entsprechende Parameter zugewiesen. Beginnend mit den beiden Pflichtwerten für die horizontale und vertikale Verschiebung, folgt wiederum die Angabe zum Grad der Unschärfe. Zuletzt wird die Farbe des Schattens definiert, welche dank CSS3 auch transparent sein kann.

```
#beispiel-1 { text-shadow: 2px 2px 0px rgba(0,0,0,1); }
```

Die Parameter zur Verschiebung erlauben negative Werte, außerdem ist die Angabe mehrerer Schatten zulässig und wird bei folgender Schreibweise korrekt interpretiert:

```
#beispiel-2 {  
    text-shadow:      2px 2px 0px rgba(255,255,255,1),  
                     4px 4px 0px rgba(0,0,0,1);  
}
```

## BEISPIEL-1    BEISPIEL-2

Abb 19: Veranschaulichung Textschatten - `#beispiel-1` und `#beispiel-2`

Auf Präfixe für separate Browser kann bei `text-shadow` verzichtet werden, da die meisten Browser mit der CSS-Regel etwas anfangen können.

---

64 Vgl. W3 Consortium: CSS Text Level 3, 14.08.2012, <http://www.w3.org/TR/css3-text/#text-shadow> (29.10.2012)

IE	FIREFOX	CHROME	SAFARI	OPERA	IOS SAFARI	OPERA MINI	ANDROID	BLACKBERRY
9.0	14.0	21.0	5.1	12.0	3.2	-	2.1	7.0

Tabelle 11: Ab welcher Version werden Schatten von Browsern unterstützt - Quelle: <http://caniuse.com>

### 3.2.4 Multiple Hintergründe

Um einen HTML-Container mit mehreren Hintergrundbildern zu versehen, hatte man bisher nur die Möglichkeit, zusätzliche für den Inhalt unnötige HTML-Container hinzuzufügen, um jede mit einem Hintergrund auszustatten. Diese Technik verstieß jedoch gegen den Grundsatz einer strikten Trennung von Inhalt und Gestaltung. Mit dieser Methode wurden HTML-Elemente einzig dafür eingesetzt, um ein bestimmtes Design zu realisieren.

Dank CSS3 kann auf verschachtelte HTML-Container verzichtet werden, da es dem Gestalter erlaubt, ein einziges Element mit mehreren Hintergrundbildern zu versehen.<sup>65</sup> Wie schon bei der Angabe multipler Textschatten ist es möglich, durch Komma getrennt die gewünschten Hintergrundbilder mitsamt Positionierungsangaben aufzuzählen. Dies sieht beispielsweise so aus:

```
#multiple-bg {  
    background: url("bild1.jpg") top left no-repeat,  
                url("bild2.jpg") top right no-repeat,  
                url("bild3.jpg") bottom right no-repeat,  
                url("bild4.jpg") bottom left no-repeat;  
}
```

Als Resultat erhält das HTML-Element `#beispiel-1` in jeder Ecke das entsprechende Bild als Hintergrund (Abbildung 20).

---

<sup>65</sup> Vgl. W3 Consortium: CSS Backgrounds and Borders Module Level 3, 24.07.2012, <http://www.w3.org/TR/css3-background/#layering> (29.10.2012)

Abb. 20: Veranschaulichung multipler Hintergrundbilder für `#multiple-bg`

Multiple Hintergründe können von fast allen Browsern korrekt dargestellt werden (Tabelle 12), sodass der Einsatz der Technik sicher ist. Lediglich ältere Versionen des *Internet Explorers* verstehen die CSS-Regel in dieser Form nicht. Hier wird lediglich das zuerst angegebene Bild interpretiert und dargestellt. Jedoch lässt sich in diesen Fällen mit JavaScript nachhelfen (Vgl. Punkt 3.3 *Neue Einsatzmöglichkeiten von JavaScript*).

IE	FIREFOX	CHROME	SAFARI	OPERA	IOS SAFARI	OPERA MINI	ANDROID	BLACKBERRY
9.0	14.0	21.0	5.1	12.0	3.2	5.0-7.0	2.1	7.0

Tabelle 12: Ab welcher Version werden multiple Hintergrundbilder von Browsern unterstützt  
Quelle: <http://caniuse.com>

### 3.2.5 Schriftarten

Dank CSS3 stehen Webdesignern nun zur Gestaltung einer Webseite nicht mehr nur die Schriftarten zur Verfügung, die auf den Computern der Benutzer installiert sind. Die bekanntesten sind *Arial* bzw. *Helvetica*, *Times New Roman* bzw. *Times* oder das allseits beliebte *Comic Sans*.<sup>66</sup> Mit CSS3 besteht nun die Möglichkeit, jede für das Design benötigte Schriftart auch zu verwenden. Schriftarten können direkt in die Webseite eingebunden werden, sodass der Seitenbesucher diese herunterlädt, sollte sie nicht auf dem Computer installiert sein.<sup>67</sup>

<sup>66</sup> Vgl. Comic Sans Criminal: You're a Comic Sans Criminal, <http://comicsanscriminal.com/> (29.10.2012)

<sup>67</sup> Vgl. W3 Consortium: CSS Fonts Module Level 3, 23.08.2012, <http://www.w3.org/TR/css3-fonts/#font-face-rule> (29.10.2012)

Man verwendet dazu die sogenannte `@font-face` Regel, mit der man neue Schriftfamilien (`font-family`) definieren und infolgedessen einsetzen kann. Benötigt wird die Schriftart als Datei in unterschiedlichen Formaten, da die jeweiligen Browser nur bestimmte davon interpretieren können. So benötigt der Internet Explorer das Dateiformat `.eot`, Safari Browser jedoch `.ttf`. Somit müssen mehrere Formate eingebunden werden, um eine browserübergreifende korrekte Schriftdarstellung zu gewährleisten.

	EOT	SVG	SVGZ	TrueType	WOFF
Internet Explorer	Ja	X	X	X	Ja
Firefox	X	X	X	Ja	Ja
Chrome	X	Ja	Ja	Ja	Ja
Opera	X	Ja	Ja	x	Ja
iPhone	X	Ja	X	X	X
iPad	X	Ja	Ja	X	X
Safari	X	Ja	Ja	Ja	X

Tabelle 13: Übersicht Browsersupport der Font-Formate

Quelle: <http://socialcompare.com/en/comparison/browser-fonts-support-comparison>

Die Definition einer neuen Schrift geschieht zu Beginn eines Stylesheets.<sup>68</sup>

```
@font-face {
  font-family: 'Neue Schrift';
  src: url('NeueSchrift.eot');
  src: url('NeueSchrift.eot?#iefix') format('embedded-opentype'),
        url('NeueSchrift.woff') format('woff'),
        url('NeueSchrift.ttf') format('truetype'),
        url('webfont.svg#svgFontName') format('svg');
}
```

Nachdem man eine Bezeichnung der Schriftart definiert hat, werden die Dateipfade für die einzelnen Schriftdateien angegeben. Die angegebene Methode wurde von der Firma *Fontspring* entwickelt, einem anerkannten Anbieter und Verkäufer von Web-

68 Vgl. Fontspring: Further Hardening of the Bulletproof Syntax,  
<http://www.fontspring.com/blog/further-hardening-of-the-bulletproof-syntax> (29.10.2012)

fonts. Sie hat sich als die von die meisten Browsern verstandene Methode herausgestellt. Ist die Schrift definiert, kann sie wie die anderen Schriftarten eingesetzt werden:

```
#beispiel-1 {
    font-family: "NeueSchrift", "Arial", "Helvetica";
}
```

Tabelle 13 zeigt, dass `@font-face` in Abhängigkeit von den jeweiligen Schriftformaten von fast allen Browsern unterstützt wird und ohne Bedenken eingesetzt werden kann. Zu Bedenken ist jedoch, dass sich die Ladezeiten der Webseite mit jeder Schriftart erhöhen. Die Größe einer Schriftdatei, unabhängig vom Format, beträgt meist mehr als 50 Kb.<sup>69</sup> Daher sollte sich der Einsatz individueller Schriften auf das nötige Minimum beschränken.

IE	FIREFOX	CHROME	SAFARI	OPERA	IOS SAFARI	OPERA MINI	ANDROID	BLACKBERRY
7.0	14.0	21.0	5.1	12.0	3.2	-	2.2	7.0

Tabelle 14: Ab welcher Version wird das `@font-face`-Attribut von Browsern verstanden  
Quelle: <http://caniuse.com>

### 3.2.6 Selektoren

Neben den neuen grafischen Gestaltungsmethoden hat sich auch im Bereich der Selektoren einiges getan. So lassen sich neben IDs und Klassen nun auch direkt Attribute via CSS ansprechen. Um einzelnen Links in einer Navigation ein gesondertes Layout zu geben, können diese nun über das `href`-Attribut direkt ausgewählt werden. Den Link `<a href="www.google.de">Google</a>` kann man so wie folgt bestimmen:

```
a[href="www.google.de"] { color: blue; }
```

<sup>69</sup> Vgl. Fontsquirrel: font-face Kits,  
<http://www.fontsquirrel.com/fontface> (29.10.2012)

Das Attribut mitsamt Parameter wird dabei in eckigen Klammern definiert. Hilfreich kann diese Methode beispielsweise auch bei der Gestaltung eines Kontaktformulars sein. Da jedes `<input>`-Feld für gewöhnlich mit einem `rel`-Attribut ausgestattet ist, kann man diese so sehr leicht ansprechen und gesondert gestalten.

CSS3 führt außerdem eine Reihe neuer Pseudo-Elemente ein, die es ebenfalls erleichtern, ganz bestimmte Elemente der Webseite gesondert anzusprechen. Neben gängigen Pseudo-Elementen wie `:before` und `:after` gibt es nun unter anderem das Element `:nth-child`, welches erlaubt, in einer Liste beispielsweise das dritte oder aber jedes vierte Element anzusprechen. Dies geschieht über einen Parameter. Beträgt dieser beispielsweise „3“, so betrifft dies lediglich das dritte Element der Liste. Hätte er jedoch den Wert „ $2n+2$ “, würden die Gestaltung auf jedes zweite Element angewendet werden, da sich „ $n$ “ immer um den Wert 1 erhöht.<sup>70</sup>

```
#beispiel-1 li:nth-child(3)    { background: red; }
#beispiel-2 li:nth-child(2n+2) { background: red; }
```

Es gibt noch eine Reihe anderer neuer Pseudo-Elemente. So spricht `:last-child` beispielsweise das letzte Element in einer Liste an. Das Element `:empty` betrifft alle Elemente, die keinen Inhalt besitzen.

IE	FIREFOX	CHROME	SAFARI	OPERA	IOS SAFARI	OPERA MINI	ANDROID	BLACKBERRY
9.0	14.0	21.0	5.1	12.0	3.2	5.0-7.0	2.1	7.0

Tabelle 15: Ab welcher Version unterstützen Browser CSS3 Selektoren - Quelle: <http://caniuse.com>

### 3.2.7 Transformationen

Ein an Bedeutung gewinnendes Feature von CSS3 ist die Möglichkeit, HTML-Elemente zu transformieren. Dabei wird zwischen einer Transformation im zweidimensiona-

<sup>70</sup> Vgl. W3 Consortium: Selectors Level 3, 29.09.2011, <http://www.w3.org/TR/selectors/> (29.10.2012)



len und im dreidimensionalen Raum unterschieden, wobei ersteres von weitaus mehr Browsern unterstützt wird. Daher soll nachfolgend nur diese Variante vorgestellt werden. CSS-Transformation stellt außerdem eine der Grundlagen für das Animieren via CSS3 dar (Vgl. 3.2.10 *Ausblick & Fazit zu CSS3*).

Früher liessen sich die nachfolgend beschriebenen Effekte ausschließlich via JavaScript erzeugen. Die CSS3-Methoden der Transformation können dabei auf jedes beliebige HTML-Element angewandt werden. Der dazugehörige CSS-Befehl lautet `transform` und benötigt für bestmögliche Browserkompatibilität wieder entsprechende Präfixe für die einzelnen Browser. Zur Verfügung stehen dabei fünf Methoden der Transformation:<sup>71</sup>

<code>translate()</code>	Das Objekt wird entsprechend der angegebenen Parameter neu positioniert. Angegeben werden die Entfernungen auf X- und Y-Achse ausgehend vom Original.
<code>rotate()</code>	Das Objekt wird um den angegebenen Parameter gedreht. Angegeben wird das Ausmaß der Drehung in Grad, wobei sich das Objekt im Uhrzeigersinn dreht. Negative Werte sind dabei trotzdem erlaubt.
<code>scale()</code>	Das Objekt wird neu skaliert, abhängig von der Höhe und Breite. Angegeben wird dabei ein Vielfaches der originalen Größe.
<code>skew()</code>	Das Objekt wird horizontal und vertikal verzerrt.
<code>matrix()</code>	Mit dieser letzten Methode lassen sich die vorherigen Transformationen kombiniert anwenden.

Angewendet wird zweidimensionale Transformation wie folgt:

```
#beispiel-translate {  
    transform: translate (20px, 20px);  
}
```

---

71 Vgl. W3 Consortium: CSS Transforms, 11.09.2012, <http://www.w3.org/TR/css3-2d-transforms/#two-dimensional-subset> (29.10.2012)

```
#beispiel-rotate {
    transform: rotate(45deg);
}
#beispiel-scale {
    transform: scale(1,2);
}
#beispiel-skew {
    transform: skew(30deg,30deg);
}
```

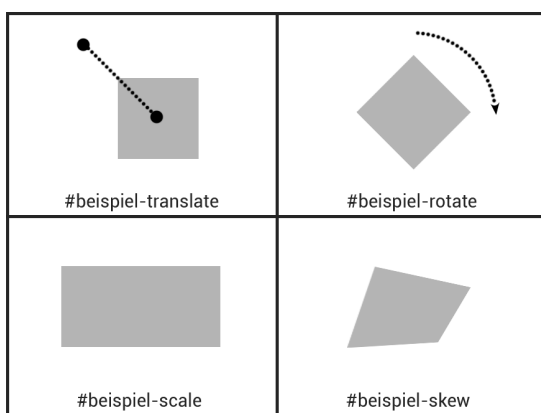


Abb. 21: Veranschaulichung der obigen Beispiele zu 2D-Transformation

IE	FIREFOX	CHROME	SAFARI	OPERA	IOS SAFARI	OPERA MINI	ANDROID	BLACKBERRY
9.0	14.0	21.0	5.1	12.0	3.2	-	2.1	7.0

Tabelle 16: Ab welcher Browserversion werden 2D Transformationen unterstützt - Quelle: <http://caniuse.com>

### 3.2.10 Ausblick & Fazit zu CSS3

Der Prozess, dass Browser mit jeder neuen Version weitere Features von CSS in ihr Repertoire integrieren und somit verstehen, wird auch in Zukunft nicht aufhören. Mittlerweile wird fast jede CSS-Regel der *Cascading Style Sheets Level 2 Revision 1* von modernen Browsern korrekt interpretiert.<sup>72</sup> Diese Entwicklung ist auch bei CSS3 zu verzeichnen, jedoch befinden wir uns dort noch in einem viel früherem Stadium. Nach

71 Vgl. Quirksmode: CSS contents and browser compatibility, <http://www.quirksmode.org/css/contents.html> (29.10.2012)

und nach erreichen einzelne CSS3-Features jedoch einen Integrationsstand, der dessen Einsatz möglich macht. So sind bereits vorgestellte Techniken wie Schlagschatten und abgerundete Ecken überall im Netz vorzufinden. Andere Funktionen werden diesen Stand erst in den kommenden Jahren erreichen, und sollen nun kurz vorgestellt werden. So wurde die bereits vorgestellte Eigenschaft `transform 2009` von *WebKit*-Entwicklern um weitere dreidimensionale Features erweitert.<sup>72</sup> Mittlerweile verstehen auch andere moderne Browser diese CSS-Befehle (Tabelle 16). Das Entwicklerteam des *Internet Explorers* arbeitet daran, diese dreidimensionalen Features in die kommenden Versionen ihrer Browser zu integrieren.<sup>73</sup> Einige der neuen Parameter für eine dreidimensionale Transformation sind.<sup>74</sup>

<code>rotateX()</code>	Wie schon bei <code>rotate()</code> wird ein Gradwert für das Ausmaß der Rotation angegeben. Die Rotation erfolgt nun jedoch um eine imaginäre horizontale Achse im dreidimensionalen Raum.
<code>rotateY()</code>	Hier erfolgt die Rotation um eine imaginäre vertikale Achse.
<code>translate3d()</code>	Wie bei <code>translate()</code> erfolgt eine Verschiebung, nun jedoch ergänzt um einen weiteren Wert, der die Verschiebung entlang der Raumachse definiert.
<code>scale3d()</code>	Wie bei <code>scale()</code> erfolgt eine Skalierung, nun jedoch ergänzt um einen weiteren Wert, der die Vergrößerung entlang der Raumachse definiert.

IE	FIREFOX	CHROME	SAFARI	OPERA	IOS SAFARI	OPERA MINI	ANDROID	BLACKBERRY
10	14.0	21.0	5.1	-	3.2	-	3	7.0

Tabelle 17: Ab welcher Browserversion werden 3D Transformationen unterstützt - Quelle: <http://caniuse.com>

Ein weiteres großes Feld ist das Animieren via CSS3, auch als Transitions bezeichnet. Die damit verbundenen Features werden in Zukunft eine immer wichtigere Rol-

72 Vgl. WebKit: 3D Transforms, 16.07.2009, <https://www.webkit.org/blog/386/3d-transforms/> (29.10.2012)

73 Vgl. Microsoft: Internet Explorer 10 - 3D Transforms, [http://msdn.microsoft.com/en-us/library/ie/hh673529\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/ie/hh673529(v=vs.85).aspx) (29.10.2012)

74 Vgl. W3 Consortium: 3D Transform Properties, 16.07.2009, <http://www.w3.org/TR/css3-3d-transforms/#transform-property> (29.10.2012)

le spielen, da sie entscheidende Vorteile mit sich bringen. So konnten Animationen jeglicher Art auf einer Webseite bisher ausschließlich mittels JavaScript realisiert werden. JavaScript-Bibliotheken wie *jQuery* oder *Mootools* haben sich für Zwecke wie diesen etabliert und kommen entsprechend oft zum Einsatz. Jedoch hat beispielsweise die aktuelle *jQuery*-Bibliothek 1.8.2 eine Größe von 92 KB.<sup>75</sup> Mittels CSS3-Transitions kann auf jegliches JavaScript verzichtet und trotzdem das gleiche Resultat erzielt werden. Außerdem zeigt der direkte Vergleich, dass CSS-Transitions gegenüber JavaScript-Animationen eine bessere Performance aufweisen.<sup>76</sup> Auf die Grundlagen der Techniken zum Animieren von HTML-Elementen soll nachfolgend kurz eingegangen werden.

Der CSS-Befehl zum animieren lautet `transition` (browserübergreifend mit entsprechenden Präfixen). Dabei soll ein HTML-Element immer von einem Stil - sei es Größe, Form oder Farbe - zu einem anderen wechseln. Dazu müssen diese beiden Zustände zuerst definiert werden:

```
#beispiel-transition {  
    width:      50px;  
    height:     50px;  
}  
#beispiel-transition:hover {  
    width:      100px;  
    height:     100px;  
}
```

In diesem Beispiel wird der zweite Zustand durch das Überfahren mit der Maus ausgelöst und erscheint unmittelbar. Um diesen Wechsel zu animieren, wird das Beispiel um eine `transition` ergänzt, in der die Dauer der Animation sowie die zu animierende Eigenschaft definiert wird, in unserem Fall die Breite und Höhe.

---

75 Vgl. jQuery: jQuery v1.8.2 [jquery.com](http://code.jquery.com/jquery-1.8.2.min.js),  
<http://code.jquery.com/jquery-1.8.2.min.js> (29.10.2012)

76 Vgl. Shaw, Brad: Why are CSS3 transitions worth using?,  
<http://css3.bradshawenterprises.com/blog/jquery-vs-css3-transitions/> (29.10.2012)

```
#beispiel-transition { transition: width 2s, height 2s; }
```

Das Ergebnis ist eine Animation wie in Abbildung 22 beschrieben. Dies ist jedoch nur die Spitze dessen, was mit CSS3-Animationen möglich ist. So lassen sich selbst komplexe Bewegungs-, Verformungs- und Verfärbungsabläufe bestehend aus mehreren Schritten ausschließlich mit CSS realisieren.

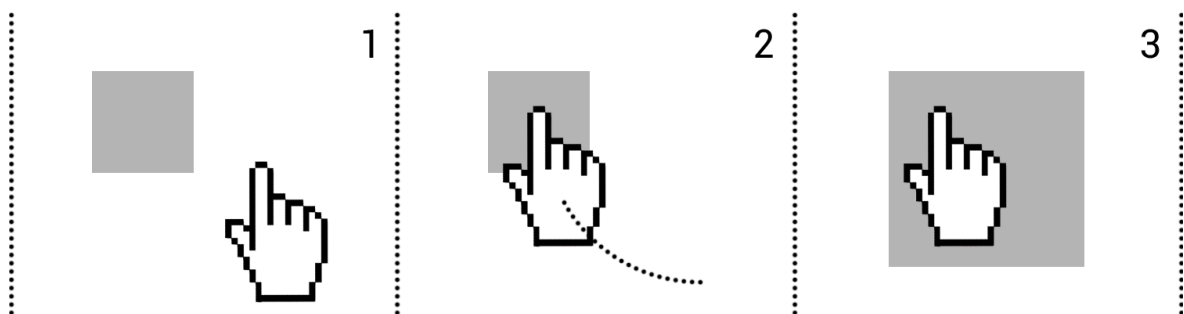


Abb. 22: Veranschaulichung der im obigen Beispiel beschriebenen CSS-Transition

IE	FIREFOX	CHROME	SAFARI	OPERA	IOS SAFARI	OPERA MINI	ANDROID	BLACKBERRY
10	14.0	21.0	5.1	12.0	3.2	-	2.1	7.0

Tabelle 18: Ab welcher Version unterstützen Browser CSS3 Transitions - Quelle: <http://caniuse.com>

Ebenso stehen neue Funktionen zur Gestaltung des Layouts in den Startlöchern, die noch mehr Freiheiten mit sich bringen und die Arbeit des Webdesigners vereinfachen. So existiert bereits eine von Browsern noch unzureichend unterstützte Funktion (Tabelle 18) zur Unterteilung eines HTML-Containers in eine beliebige Anzahl von Spalten.<sup>77</sup> So lässt sich über die CSS-Regeln `column-count` und `column-width` die Anzahl und entsprechende Breite der Spalten festlegen. Das Ergebnis ist eine einzige mehrspaltige Box, für die man sonst mehrere HTML-Container hätte einfügen müssen. Mit weiteren CSS-Regeln wie `column-code` lässt sich das Layout dabei noch weiter verfeinern, in diesem Fall eine Trennlinie zwischen den Spalten bestimmen.

<sup>77</sup> Vgl. W3 Consortium: CSS Multi-column Layout Module, 12.04.2011, <http://www.w3.org/TR/css3-multicol/> (29.10.2012)

IE	FIREFOX	CHROME	SAFARI	OPERA	IOS SAFARI	OPERA MINI	ANDROID	BLACKBERRY
10	14.0	21.0	5.1	12.0	3.2	-	2.1	7.0

Tabelle 19: Ab welcher Version wird ein multiple-column-Layout von Browsern unterstützt

Quelle: <http://caniuse.com>

Ebenso spannend ist das Konzept der flexiblen Box, kurz *Flexbox*, welches ein großes Problem bisheriger HTML-Container löst und für das Konzept des Responsive Web-designs eine große Bereicherung darstellen - wenn die Mehrzahl der Browser diese in naher Zukunft unterstützen werden (Tabelle 20).

Definiert man für ein HTML-Element entsprechend des bisherigen Boxmodels eine Breite von beispielsweise 100 Pixeln sowie einen Rahmen von 25 Pixeln, so hat das dargestellte Element am Ende eine Gesamtbreite von 150 Pixeln. Diese Addition der Rahmenbreite auf die eigentliche Breite des Containers ist in einigen Fällen störend und kann zu Problemen führen. Insbesondere bei der Kombination von Prozent- und Pixelangaben wird es schwierig, die exakt gewünschte Breite des Containers zu realisieren. Ein Container mit der Breite 50 Prozent und einem Rahmen von 25 Pixeln hat letztendlich immer eine Breite von 50 Prozent plus den 25 Pixeln an jeder Seite.

Das Flexbox-Modell löst dieses Problem.<sup>78</sup> So kann die CSS-Eigenschaft `display` nun den Wert `flex` annehmen, womit sich unter anderem eine von Rahmen unabhängige Container-Breite festlegen lässt. Außerdem lässt sich mit dem Flexbox-Modell eine Reihenfolge festlegen, mit welcher HTML-Elemente im Falle einer Auflistung untereinander (z.B. auf mobilen Geräten) erscheinen. Die Flexbox wird also in der Zukunft ein mächtiges Instrument in der Gestaltung plattformübergreifender Webseiten werden.

IE	FIREFOX	CHROME	SAFARI	OPERA	IOS SAFARI	OPERA MINI	ANDROID	BLACKBERRY
10	14.0	21.0	5.1	12.1	3.2	-	2.1	7.0

Tabelle 20: Ab welcher Version wird die *Flexbox* von Browsern unterstützt - Quelle: <http://caniuse.com>

Integrieren Browser auf der einen Seite mit jeder neuen Version weitere CSS-Features,

<sup>78</sup> Vgl. W3 Consortium: CSS Flexible Box Layout Module, 18.10.2012, <http://www.w3.org/TR/css3-flexbox/> (29.10.2012)

so wird CSS3 auf der anderen Seite dabei jedoch ständig um neue Funktionen erweitert und an die Bedürfnisse des Webs angepasst. Ein CSS4 wird es nach heutigem Stand nicht geben. *Tab Atkins Jr.*, Mitglied der *CSS Working Group* hat klargestellt, dass heute und auch in Zukunft kein CSS4 geplant ist.<sup>79</sup> Der Begriff CSS3 beschreibt somit alle nach CSS2.1 bereits veröffentlichten sowie kommende Weiterentwicklungen der *Cascading Style Sheets*. Zukünftig werden die einzelnen Module separat weiterentwickelt.

### 3.3 Neue Einsatzmöglichkeiten von Javascript

Setzt man moderne Webstandards wie HTML5 und CSS3 auf seiner Webseite ein und will Benutzer älterer Browser trotzdem das gleiche Surferlebnis bieten, kommt man um den Einsatz von JavaScript nicht herum. Nachfolgend sollen daher sehr nützliche Hilfsmittel vorgestellt werden, mit denen dieses Ziel erreicht werden kann.

#### 3.3.1 JavaScript-Bibliotheken

Möglich wird der bedenkenlose Einsatz von HTML5 und CSS3 erst durch spezielle JavaScript-Bibliotheken wie *Modernizr* ([www.modernizr.com](http://www.modernizr.com)), dessen Ziel es ist, Browser, die die modernen Features nicht interpretieren können, mit alternativen JavaScript-Lösungen zu versorgen. So soll eine Webseite auf wirklich jedem Browser das selbe Erscheinungsbild und selben Funktionsspektrum haben. *Modernizr* untersucht nach dem fertigen Laden der Webseite innerhalb weniger Millisekunden, um welchen Browser es sich handelt, welche Features dieser nicht unterstützt und ob diese auf der Seite zum Einsatz kommen.

Es unterstützt dabei HTML5-Features wie Canvas, Drag&Drop oder auch Audio- und Videoelemente.<sup>80</sup> Auch viele der bereits beschriebenen CSS3-Funktionen werden

---

79 Vgl. Atkins, Tab Jr.: There is no such thing as CSS4, 05.09.2012, <http://www.xanthir.com/b4Ko0> (29.10.2012)

80 Vgl. Modernizr: Modernizr Features, <http://modernizr.com/docs/#features-html5> (29.10.2012)

durch den Einsatz von *Modernizr* auch auf älteren Browsern korrekt gerendert. Das betrifft beispielsweise die CSS-Regeln `@font-face`, `border-image`, `border-radius`, `text-shadow` oder auch das bereits vorgestellte *Flexbox*-Modell. Mittlerweile ist das Plugin zu einem beliebten und sehr populären Hilfsmittel in der Webentwicklung avanciert. Große Portale wie *Twitter*, *Google* oder *Microsoft* setzen es bereits auf ihren Webseiten ein.

*Modernizr* ist jedoch nicht die einzige JavaScript-Bibliothek, die einen browserübergreifenden Einsatz von HTML5- und CSS3-Funktionen ermöglicht. So macht das Plugin *selectivizr* ([www.selectivizr.com](http://www.selectivizr.com)) veralteten Internet Explorer-Versionen Pseudo-Klassen verständlich. Mit *CSSPIE* ([www.css3pie.com](http://www.css3pie.com)) lassen sich CSS3-Features wie Farbverläufe, multiple Hintergründe oder Schlagschatten selbst auf dem *Internet Explorer 6* darstellen. *320 and Up* ([www.stuffandnonsense.co.uk/projects/320andup](http://www.stuffandnonsense.co.uk/projects/320andup)) ist ein schlankes Framework, optimiert für die Erstellung plattformübergreifender Webseiten. Auch die populären JavaScript-Bibliotheken *jQuery* und *Mootools* haben mittlerweile eine Version ihres Scripts veröffentlicht, die auch mobile Endgeräte und die Zusammenarbeit mit HTML5 optimiert wurde.

### 3.3.2 CSS-Extensions

Eine weitere spannende Einsatzmöglichkeit von JavaScript hat sich in Form verschiedener CSS Erweiterungen herauskristalisiert, die die Arbeit mit der Stilsprache enorm vereinfachen. Eines der populärsten Plugins ist *{less}* ([www.less.org](http://www.less.org)), welches CSS um diverse praktische Funktionen erweitert. *{less}* erlaubt beispielsweise die Deklaration und Nutzung von Variablen im CSS-Code. So können bestimmte, auf der Webseite häufig wiederkehrende Farbwerte oder Abstände zu Beginn des Stylesheets als Variablen definiert werden.

```
h1    { color: #F326D1; }  
h2    { color: #F326D1; }  
h3    { color: #F326D1; }
```

Mit *{less}* muss der exakte Farbwert nun nur noch einmal definiert werden:



```
@color: #F326D1;
h1    { color: @color; }
h2    { color: @color; }
h3    { color: @color; }
```

Diese wie auch weitere neue Verhaltensweisen des durch *{less}* erweiterten CSS sind an Programmiersprachen angelehnt. Beispielsweise sind nun auch Verschachtelungen wie die Folgende möglich, wenn CSS-Regeln nur auf bestimmte Elemente innerhalb eines Containers zutreffen sollen. Das Beispiel

```
#header h1 { color: red; }
#header p  { font-size: 12px; }
#header p a { font-weight: bold; }
```

kann nun ebenfalls in folgender Form ausgedrückt werden:

```
#header {
  h1    { color: red; }}
  p     { font-size: 12px;
    a    { font-weight: bold; }
  }
}
```

Ein ähnliches Plugin, welches ebenfalls CSS-Features wie Variablen oder Verschachtelungen ermöglicht, ist *Sass* ([www.sass-lang.com](http://www.sass-lang.com)).

## 4 Responsive Webdesign

Die zunehmende Verbreitung mobiler Endgeräte zeigt: eine erfolgreiche Webseite muss auf stationären Computern sowie auf Smartphones und Tablets gleichermaßen gut laufen. Studien zeigen, dass mobil-optimiertes Webdesign den mobilen Traffic und die Verkaufszahlen von E-Commerce Seiten signifikant erhöht.<sup>81</sup> Betrachtet man das Leistungsspektrum renommierter Webagenturen, so ist die Optimierung auf mobile Geräte ein wichtiges Verkaufsargument geworden. Ein statisches Webseiten-Layout reicht also nicht mehr aus. Webentwickler haben jedoch Wege gefunden, sich an die neue Situation anzupassen. Es findet ein Umdenken statt: weg von festen, statischen Breiten und Höhen hin zu flexiblen Größen. Dieser Denkansatz wird auch als *Responsive Webdesign* bezeichnet. Der Begriff wurde im Zusammenhang mit plattformübergreifender Webentwicklung erstmals von *Ethan Marcotte* in einem Artikel vom 25. Mai 2010 geprägt.<sup>82</sup> Das Konzept des *Responsive Webdesign* befindet sich seit dem auf einem Erfolgskurs und wird stetig weiterentwickelt. In den folgenden Kapiteln soll die Technik dahinter detailliert vorgestellt werden.

### 4.1 Definition und Konzept

Responsive Webdesign vertritt den Ansatz, dass sich Design und die Funktionalität einer Webseite nach dem Nutzer und der technischen Umgebung richten soll. Damit ist nicht ausschließlich die Anpassung des Layouts an das vorherrschende Betrachtungsgerät gemeint, sondern jede Methode und Idee, die auf ein plattformunabhängiges Surferlebnis abzielt - sowohl in konzeptioneller als auch in technischer Hinsicht. *Responsive Webdesign* bedeutet nicht, eine Webseite für moderne Smartphones oder die gängigen Tablets zu optimieren. Das Ziel ist eher, eine Seite so zu konzipieren, dass es keine Rolle mehr spielt, welches Endgerät benutzt wird. Der Mobil-Markt ist mitt-

---

81 Vgl. Blast Advanced Media: Responsive Web Design Increases Sales, <http://www.blastam.com/assets/pdf/blast-thinktankphoto-responsive-casestudy.pdf> (29.10.2012)

82 Vgl. Marcotte, Ethan: Resonsive Webdesign, 25.05.2010, <http://www.alistapart.com/articles/responsive-web-design/> (29.10.2012)

lerweise unüberschaubar geworden und selbst ältere Feature-Phones mit kleinen Displays können auf das Internet zugreifen. *Responsive Webdesign* bedeutet: Niemand soll ausgeschlossen werden.

Das Hauptmerkmal einer responsiven Webseite ist seine flexible Architektur. An Stelle fester Breiten werden nun prozentuale Werte benutzt. Das betrifft nicht nur die Abmessungen der HTML-Container, sondern auf deren Rahmen, die Größe von Bildern und sogar Schriftgrade. Am besten lässt sich dieses Konzept an einem Beispiel verdeutlichen.

## 4.2 Analyse eines adaptiven Layouts

Das Online-Newsportal des *Boston Globe* ([www.bostonglobe.com](http://www.bostonglobe.com)) hat sein Design kürzlich auf ein responsives Layout umgestellt. Auf großen Bildschirmen erreicht das Layout dabei seine maximale Breite von 1230 Pixeln (Abbildung 23). Ein für News-Portale vergleichsweise sehr breites Layout, da es auf große Bannerwerbung im Randbereich verzichtet und lediglich kleinere Werbeflächen innerhalb der Spalten platziert werden. Ein weiterer Vorteil einer adaptiven Gestaltung ist es somit auch, große Monitorauflösungen besser ausnutzen zu können.



Abb. 23: Screenshot der *Boston Globe*-Seite auf einem großen Monitor

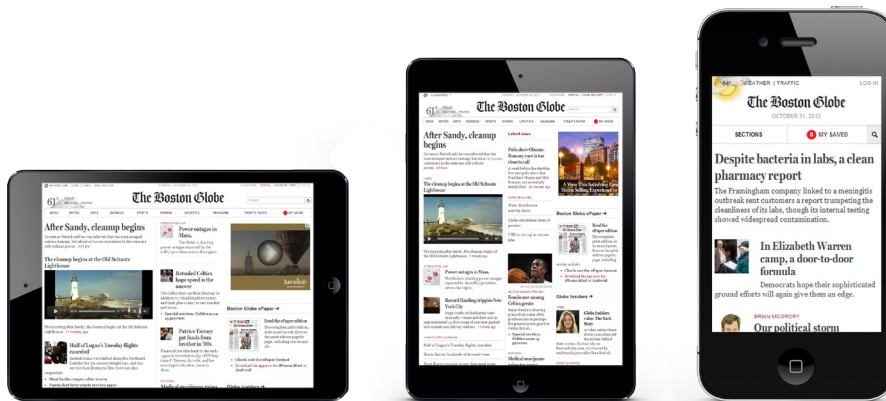


Abb. 24: Screenshot der *Boston Globe*-Seite einem Tablet quer und hochkant, sowie auf dem Smartphone

Es wird deutlich, dass die Seite auf jedem Gerät ein anderes Erscheinungsbild annimmt. Zwar bleibt der grundlegende, gestalterische Charakter bestehen, jedoch findet immer eine Optimierung statt, zum Beispiel in der Anordnung der Inhalte. Zur Veranschaulichung ist die Seite in Abbildung 25 nochmals skizziert und die unterschiedlichen Inhaltsbereiche farblich hervorgehoben. Der Header enthält das Logo und eine komplexe Navigation. Darunter befindet sich ein dreispaltiges Layout. Die linke Hauptspalte nimmt dabei 40 Prozent der Gesamtbreite ein und enthält die Top-News des Tages. Die mittlere Spalte ist mit „Latest news“ titulierte und enthält besonders aktuelle Meldungen. Sie hat eine Breite von ungefähr 25 Prozent des Gesamtlayouts. Die rechte Spalte ist 30 Prozent breit und enthält Werbung, zum einen auf eigene Produkte, wie das Boston Globe ePaper, sowie von externen Anbietern. Unterhalb dieser drei Spalten setzt sich das Layout zweispaltig fort. Die linke Spalte hat eine Breite von zwei Dritteln und enthält Meldungen der einzelnen Ressorts. Die rechte Spalte enthält verschiedene andere Kategorien.



Abb. 25: Layoutskizzierung der *Boston Globe*-Seite

Wird der Browser nun kontinuierlich verkleinert, geschehen verschiedene Dinge mit dem Layout. Zum einen passt sich die Gesamtbreite immer der Browserbreite an, wobei links und rechts ein minimaler weißer Rand bestehen bleibt. Die einzelnen Bereiche müssen sich dadurch nun ebenfalls anpassen. So verringern sich die Abstände zwischen den einzelnen Navigationspunkten und das Suchformular wird kleiner. Die drei Spalten im oberen Bereich verkleinern sich ebenfalls, wobei sich das Seitenverhältnis 40:25:35 nicht verändert. Als Folge rückt der Text enger zusammen, Bilder und Videos werden kleiner und passen sich an. Ähnliches geschieht im zweispaltigen Bereich.



Abb. 26: Veranschaulichung des Layoutverhaltens bei kleiner werdender Browserbreite

Unterschreitet die Browserbreite 800 Pixel, findet ein Wechsel des Layouts statt. In einer dreispaltigen Anordnung würden die Spalten nun zu schmal werden und einen gestalterischen Mangel darstellen. Die Lösung ist eine Reduktion der Spalten hin zu einer inhaltlichen Anordnung untereinander - ein Hauptmerkmal des *Responsive Webdesign*. Dadurch ist der *Boston Globe* nun gezwungen, seinen Inhalte der Wichtigkeit nach zu sortieren. So rutscht die rechte Spalte, die die Werbung zu den eigenen Produkten enthielt, unter die beiden anderen. Der Werbebanner wird jedoch in der neuen rechten Spalte oben eingefügt. Es bleibt ein zweispaltiges Layout mit einem Breitenverhältnis von 1:1. Auch unten rutscht die rechte Spalte unter die breitere Linke, welche nun die gesamte Breite ausfüllt. (Abbildung 27)

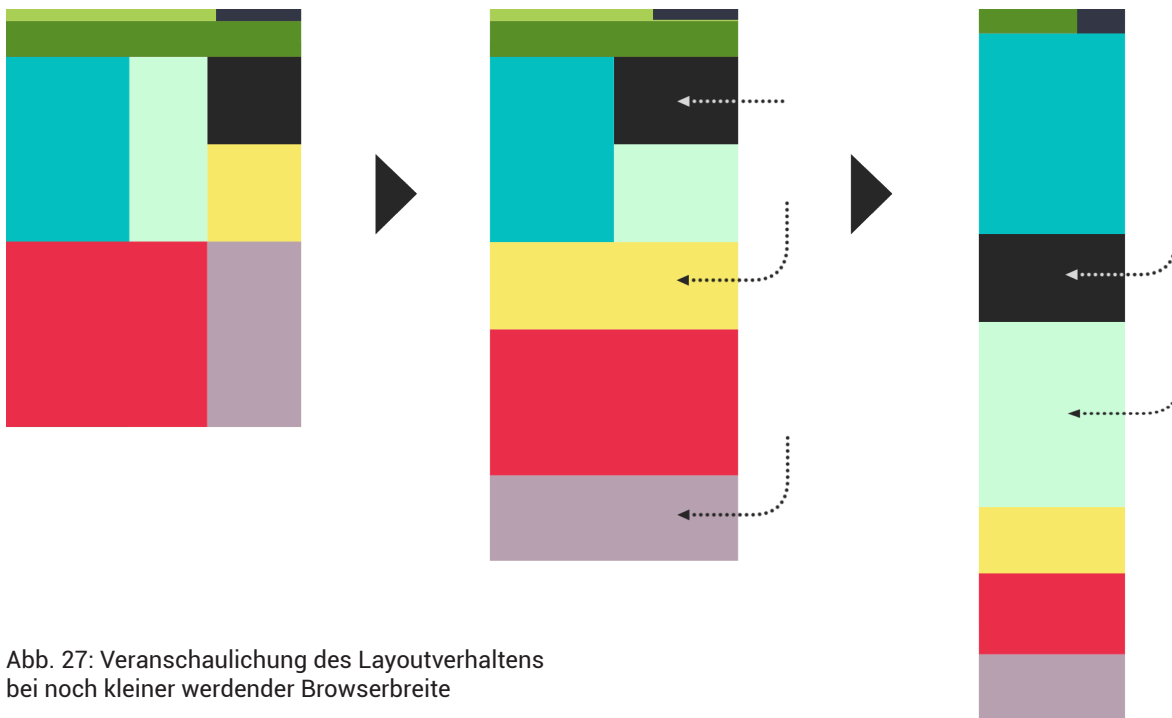


Abb. 27: Veranschaulichung des Layoutverhaltens bei noch kleiner werdender Browserbreite

Ein weiterer Wechseln findet in der Navigation statt. Für das ausladende Menu mit den einzelnen Ressorts reicht der Platz nun nicht mehr aus. Diese werden nun in einem Drop-Down-Menu angeordnet, dass durch einen einzigen Button geöffnet werden kann. Durch diese platzsparende Lösung kann das Suchfeld nun ebenfalls rechts neben dem Menu positioniert werden (Abbildung 28)



Abb. 28: Reduktion der Navigation

Ab einer Breite von 640 Pixeln wechselt man zu einem komplett einspaltigen Layout (Abbildung 27). Nun werden alle Inhalte untereinander angeordnet - die optimalste Darstellung auf mobilen Telefonen. Der *Boston Globe* hat sich dazu entschieden, die

Top-News nun an erster Stelle zu platzieren, gefolgt von den „Latest News“ der ehemals rechten Spalte. Großen Überschriften werden nun kleinere Schriftgrade zugewiesen. Auch das Logo wird konstant herunterskaliert und statt des Suchfeldes wird nur noch ein Lupen-Icon angezeigt. Jede dieser Anpassungen ist eine platzsparende Maßnahme, um den Inhalten die größtmöglichen Fokus zu geben und den kleinen Bildschirm optimal ausnutzen zu können.

## 4.3 Die Alternative: Separate mobile Webseiten

Bevor auf die Konzeption eines responsiven Layouts eingegangen wird, soll kurz auf die Technik der separaten mobile Webseiten eingegangen werden. Gemeint ist damit eine automatische Weiterleitung auf eine eigenständige, für mobile Webseiten optimierte Version, sollte der Nutzer von einem mobilen Endgerät aus die Seite besuchen. Somit stellen sie eine Alternative zum *Responsive Webdesign* dar.

Oftmals wird das Konzept angewandt, wenn eine komplette Umstellung der Hauptseite zu anwändig ist, oder man auf der mobilen Version nur einen speziellen Service anbieten will. Viele News-Portale wie *Spiegel Online* oder Shops wie *Amazon* nutzen diese Methode. Man kann die mobile Seite separat von der Hauptseite gestalten und ist nicht gezwungen, alle Inhalte der Desktop-Version zu übernehmen. Die mobile Seite folgt dabei jedoch den selben Prinzipien des *Responsive Webdesign*, sodass die folgenden Kapitel auch für diese Variante relevant sind.

Methoden zur Weiterleitung gibt es viele. Mittels JavaScript kann beispielsweise beim Seitenaufruf überprüft werden, um welchen Browser es sich handelt. Das Problem besteht jedoch darin, dass angesichts der Vielzahl an Browsern nicht korrekt weitergeleitet wird. Daher werden auf der Seite meist entsprechende Menüpunkte eingefügt, um zwischen beiden Versionen hin- und herwechseln zu können. Die mobile Seite hat dabei meist eine URL in der Form `mobil.seitenname.de` oder `m.seitenname.de`.



## 4.4 Konzeption responsiver Webseiten

Der Gestaltungsprozess einer Webseite hat sich mit dem Aufkommen des *Responsive Webdesign* stark geändert. Unter anderem ist er mit mehr Aufwand verbunden, da man nicht mehr nur für Monitore mit einem großen Bildschirm innerhalb eines festen Rasters, sondern für unterschiedlichste Endgeräte entwickelt. Diese Faktoren müssen von Beginn an in den Entwicklungsprozess der Webseite einfließen. Bevor es um die genaue technische Umsetzung eines adaptiven Layouts geht, soll zuvor auf effektive Strategien sowie verschiedene relevante Aspekte bei der Konzeption responsiver Webseiten eingegangen werden.

Das Beispiel des *Boston Globe* hat gezeigt, dass es erforderlich ist, alle Inhalte der Seite nach seiner Wichtigkeit zu bewerten. Kleine Bildschirmen zwingen dazu, den Content untereinander anzuordnen. Daraus schlussfolgernd ist es nur effektiv, wenn die Seite zuerst für mobile Geräte konzipiert wird. Somit erfolgt die inhaltliche Priorisierung gleich zu Beginn und man muss sich darüber klar werden, welche Elemente auf die Seite sollen und wie wichtig diese sind. Der kleine Bildschirm zwingt, sich auf das Wesentliche zu konzentrieren.

Das Konzept wurde erstmals von *Luke Wroblewski* in dessen Buch *Mobile First* vorgestellt<sup>83</sup> und beschreibt die Arbeitsweise, Webseiten von der kleinsten hin zur größten Layout-Version zu entwickeln. Bisher werden die kleinen Bildschirme mobiler Geräte unter anderem mit negativen Eigenschaften wie Platzmangel verbunden. *Mobile First* schlägt dagegen vor, die vermeintliche Schwäche als Vorteil zu betrachten und durch Verzicht auf unwesentliche Informationen und Bedienelemente ein benutzerfreundliche, übersichtliche und optisch angenehm reduzierte Designlösung für alle Bildschirmgrößen zu finden. Neben der inhaltlichen wird man somit auch zu einer gestalterischen Fokussierung gezwungen. Zum einen erlaubt der kleine Bildschirm kein ausschweifendes Layout und zum anderen ist dieses meist auch gar nicht erstrebenswert. Oftmals ist es sogar schwieriger, ein grafiklastiges Layout nachträglich für mobile Geräte zu optimieren, als anders herum.

---

83 Vgl. Wroblewski, Luke: *Mobile First*, 19.01.2012, <http://www.abookapart.com/products/mobile-first> (29.10.2012)



Bei der bereits beschriebenen inhaltlichen Priorisierung sollte ein weiterer Aspekt beachtet werden. Die entscheidende Frage lautet hier, ob PC-Nutzer die gleichen Ansprüche an eine Webseite haben als Nutzer mobiler Geräte. Muss das mobile Layout die gleichen Inhalte liefern wie die Desktop-Version? Besucht man die Seite einer Kaufhausfiliale via Smartphone, ist man möglicherweise eher daran interessiert, die Adresse der nächstmöglichen Filiale zu erfahren als den Online-Shop zu nutzen. So könnte auf der mobilen Seite der Shop ausgeblendet und die Adressdaten an die erste Stelle gerückt werden. Bisher ist es jedoch schwierig, den Kontext eines Seitenbesuches zu ermitteln. Eine mögliche Methode wäre die Nutzung der bereits vorgestellten HTML5 Geolocation-API, um den Standort des Benutzers (von unterwegs oder zu Hause aus) zu ermitteln und so auf den Zweck des Seitenaufrufs zu schließen. Studien des Online-Shops *Orange World* zeigten jedoch, dass die Einkäufe via Smartphone kaum vom jeweiligen Ort abhängen.<sup>84</sup> Der mobile Kontext lässt sich demnach bisher nur vermuten. Sollte man die mobile Seite auf eine bestimmte Funktion auslegen, so ist ein Link zu den anderen Angeboten dringend erforderlich.

Für die Formatierung des Inhalts hat man weitaus mehr Möglichkeiten als den Wechsel vom mehrspaltigen Layout zur Anordnung untereinander. Im Hinblick auf die Bildschirmunterschiede kann es nützlich sein, Inhalte platzsparend in Akkordeon-Form oder in Slideshows via JavaScript zusammenzufassen (Abbildung 29) - oder aber verzichtbare Informationen ganz auszublenden. Elemente wie Bilderslideshows sollten an das Betrachtungsgerät angepasst werden. Macht auf einem Tablet die Anordnung von drei Bildern nebeneinander Sinn, so ist auf Smartphonebildschirmen möglicherweise nur ein Bild optimal.

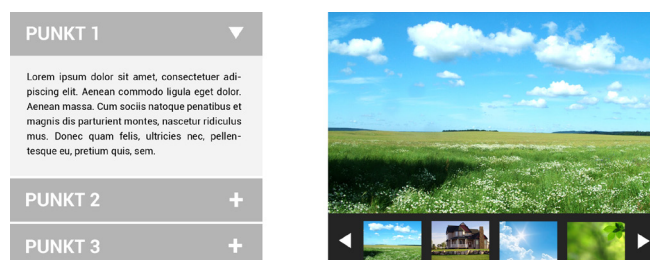


Abb. 29: Beispiel für ein Akkordeon Element zum Ausklappen von Inhalten, sowie eine Bilderslideshow

84 Vgl. Orange World: Mobile shopping is popular when watching TV, says Orange UK research, 16.12.2010, <http://thefonecast.com/Home/tabid/61/ArticleID/3602/ArtMID/538/Default.aspx> (29.10.2012)

Ebenfalls am Beispiel des *Boston Globe* deutlich geworden ist die spezielle Verwendung von Maßeinheiten. Kamen prozentuale Abstände auf älteren Seiten nur selten zum Einsatz, basiert nun ein Großteil des Layouts auf ihnen. Es müssen optimale Seitenverhältnisse der einzelnen Bereiche gefunden, aber auch die minimalen und maximalen Grenzwerte definiert werden. Die Einhaltung eines bestimmten Spaltenverhältnisses macht immer nur bis zu einem bestimmten Grad Sinn. Sollte die Seite grafische Elemente wie einen Banner zur Gestaltung nutzen, muss dieser so konzeptioniert werden, dass er sich an das Design anpasst. Er kann sich entsprechend der Breite des Layout skalieren oder abgeschnitten werden.

Sehr genau sollte die Gestaltung der Navigation erfolgen. Enthält diese viele Menüpunkte, so muss gut überlegt werden, wie die Links angeordnet werden. Fast immer führt eine Auflistung unter- oder nebeneinander auf kleinen Geräten zu Platzmangel, so dass hier eine Umkonzeptionierung erfolgen muss. Das Beispiel *Boston Globe* hat eine Lösung präsentiert, die sich als sehr praktisch erwiesen hat und mittlerweile äußerst populär ist. So wird das gesamte Navigationsmenu ausgeblendet und durch einen Button ersetzt, über den dieses aufgerufen werden kann (Abbildung 28).

Angesichts der vielen Aspekte, die bei der Konzeption zu beachten sind, wird deutlich, dass *Responsive Webdesign* mit mehr Aufwand verbunden ist. Es erfordert von Beginn an detaillierte Überlegungen zum Aufbau von Layout und Inhalt. Auch wenn die Idee lautet, eine Seite für alle Endgeräte zu entwickeln, so sieht die Realität letztlich so aus, dass man mehrere Seiten parallel entwickelt. Das Konzept sieht zwar eine fließende Layoutanpassung vor, so sind es am Ende trotzdem die drei modernen Grundpfeiler Smartphone, Tablet und PC, für die optimiert wird.

Steht das Konzept der Webseite, beginnt die technische Umsetzung. Ein gutes Konzept erleichtert dabei die Arbeit enorm. Jedoch muss auch hier ebenso gut vorausgeplant werden. Die nächsten Kapitel werden beschreiben, wie die Idee *Responsive Webdesign* technisch dank moderner Webtechnologien realisiert werden kann.

## 4.5 Technische Umsetzung

### 4.5.1 Media Queries

Media Queries sind der Schlüssel zu erfolgreichem *Responsive Webdesign*, denn mit ihrer Hilfe ist es möglich, separate CSS-Regeln bei bestimmten Geräteeigenschaften wie der Breite des Bildschirms festzulegen. Media Queries sind ein Teil von CSS, jedoch kein neues Feature. Bereits in der CSS2 Recommendation gab es die Möglichkeit, Stylesheets für verschiedene Ausgabemedien wie Monitor oder Print zu definieren. Mit CSS3 bekommen diese jedoch nun eine Reihe neuer Funktionen. So kann nun auch auf Eigenschaften wie das Höhe-Breite-Verhältnis oder die Farbtiefe des Bildschirms zugegriffen werden.<sup>85</sup>

Media Queries können innerhalb eines Stylesheets, aber auch direkt im HTML-Code verwendet werden. Möchte man beispielsweise jeweils ein Stylesheet für eine Browserbreite unter und über 800 Pixeln einbinden, geschieht dies im <head>-Bereich des HTML-Dokumentes wie folgt:

```
<link rel="stylesheet" type="text/css" media="screen and (max-width: 800px)" href="Style1.css" />
<link rel="stylesheet" type="text/css" media="screen and (min-width: 801px)" href="Style2.css" />
```

Möchte man das Stylesheet nur für eine bestimmte Breitenspanne einsetzen, sind auch **Ausdrücke wie** `media="screen and (min-device-width: 480px) and (max-device-width: 800px)"` zulässig. Eine ähnliche Syntax kann innerhalb eines Stylesheets angewendet werden:

```
@media screen and (max-width: 800px) { h1 { color: red; } }
```

---

85 Vgl. W3 Consortium: Media Queries, 19.06.2012, <http://www.w3.org/TR/css3-mediaqueries/> (29.10.2012)

```
@media screen and (min-width:801px) { h1 { color: blue; } }
```

Es wird offensichtlich, wie diese Funktionen zur Realisation eines adaptiven Layouts eingesetzt werden können. Die Layoutwechsel auf der Seite des *Boston Globe*, wenn beispielsweise aus drei Spalten zwei werden, lassen sich so leicht realisieren. Sinkt die Browserbreite auf einen bestimmten Wert, wird der entsprechenden Spalte einfach das `float`-Attribut entzogen, so dass es unter die anderen Spalten rutscht. Gleichzeitig wird für diese ein neues Seitenverhältnis bestimmt. Auch Bildschirme mit höherer Pixeldichte wie das Retina-Display des *iPhones* lassen sich so über Media Queries exakt ansprechen. Damit können bei Bedarf für diese Bildschirme größere Grafiken eingebunden werden.<sup>86</sup>

```
@media screen and (-webkit-min-device-pixel-ratio: 2),  
screen and (max--moz-device-pixel-ratio: 2) {  
    // CSS  
}
```

Viele mobile Browser älterer Smartphones können Media Queries nicht interpretieren, daher setzt sich das bereits vorgestellte Konzept *Mobile First* bei der technischen Umsetzung der Seite fort. So ist es nicht das Stylesheet für die mobile Webseitenversion, welches über Media Queries eingebunden werden sollte, sondern das der Desktop-Version. Nur so kann sicher gestellt werden, dass jedes mobile Gerät die Seite korrekt darstellen kann. *Mobile First* hat auch bei der weiteren Umsetzung des HTML-Dokuments Vorteile. So weiß man bereits, in welcher Reihenfolge die Inhalte strukturiert werden sollen. Sollte man verschiedene Stylesheets einbinden wollen, ohne Media Queries zu benutzen, gibt es Alternativen über jQuery.<sup>87</sup>

---

86 Vgl. Bacon, Gary: Responsive Design for Retina Displays on iPad and iPhone, 16.05.2012, <http://garybacon.com/post/responsive-design-for-retina-displays-on-ipad-and-iphone/> (29.10.2012)

87 Vgl. Knight, Kayla: Responsive Web Design: What It Is and How To Use It, 12.02.2011, <http://coding.smashingmagazine.com/2011/01/12/guidelines-for-responsive-web-design/> (29.10.2012)

## 4.5.2 Viewport

Die Angabe des `<viewport>`-Metatags im `<head>`-Bereich ist unentbehrlich, um das gewünschte adaptive Layout umzusetzen. Ohne dieses öffnet ein mobiler Browser die Seite in herausgezoomten Format, so dass man individuell bestimmte Bereiche der Seite vergrößern und betrachten kann. Eventuell festgelegte CSS-Regeln für diese bestimmte Bildschirmbreite würden so nicht umgesetzt werden. Mit der Angabe des `<viewport>`-Metatags

```
<meta name="viewport" content="width=device-width" />
```

wird dem Browser mitgeteilt, dass er eine Breite entsprechend der Bildschirmabmessung zur Darstellung der Seite anwenden soll.<sup>88</sup>

## 4.5.3 Realisierung flexibler Breiten

Die Flexibilität des Layouts kann mittels verschiedener CSS-Techniken erreicht werden. Auf Maßeinheiten im Pixelformat sollte verzichtet werden. Für die Gestaltung der entsprechenden HTML-Elemente hat sich dabei eine Kombination aus `width`, `min-width` und `max-width` als nützlich erwiesen. Als Beispiel soll ein Element `<section id="wrapper">` dienen, welches eine Webseite vollständig umschließt und mit diesem somit die Gesamtbreite des Layouts festgelegt werden kann. Zur Festlegung einer Gesamtbreite, wenn die Browserbreite ausreichend groß ist und das Layout nicht beeinflusst, kann das `width`-Attribut genutzt werden.

```
#wrapper {  
    width: 1000px;  
}
```

---

88 Vgl. W3 Consortium: Mobile Web Application Best Practices, 14.12.2010, <http://www.w3.org/TR/mwabp/#bp-viewport> (29.10.2012)

Nun soll das Element seine Breite jedoch nicht beibehalten, wenn die Browserabmessungen die 1000 Pixel unterschreiten. Dabei hilft die Ergänzung `max-width: 90%;`. Sie legt die Bedingung fest, dass das Element nur dann seine volle Breite besitzen darf, wenn diese gleichzeitig nicht mehr als 90 Prozent der aktuellen Browserbreite entspricht. Ist dies nicht der Fall, passt sich das Element entsprechend an.

`min-width` kann beispielsweise bei Spalten nützlich sein. Soll eine Spalte aufgrund der Leserlichkeit eine bestimmte Breite nicht unterschreiten, kann diese mittels `min-width`-Angabe im Pixelformat festgelegt werden. Zu beachten ist bei der Verwendung von Rahmen das in Punkt 3.2.10 *Ausblick & Fazit zu CSS3* im Zusammenhang mit der Flexbox beschriebene Problem der absoluten Breiten. Hier kann entweder eine entsprechende *Flexbox* einsetzen, oder sich mit der Angabe prozentualer Rahmen behelfen. Im folgenden Beispiel besitzt das Element einen Rahmen und trotzdem eine exakte Breite von 50 Prozent:

```
#beispiel {  
    width: 46%;  
    padding-left: 2%;  
    padding-right: 2%;  
}
```

Auch Bilder können über die CSS-Regel `max-width: 100%;` flexiblen Charakter bekommen. Überschreitet das umschließende HTML-Element in seiner Breite nicht die Abmessungen des Bildes, so entspricht dieses immer der Breite des Elements. Zu beachten ist hier jedoch, dass sich die Größe des Bildes nicht verringert, sollte es verkleinert dargestellt werden. So lädt man auf mobilen Webseiten oftmals unnötig große Bilder herunter, die letztlich nur als Bruchteil ihrer eigentlichen Größe dargestellt werden. An einer Lösung für dieses Problem wird jedoch gearbeitet. So hat die WHATWG vor kurzem ihrer HTML-Spezifikation das Attribut `srcset` hinzugefügt. Die Idee besteht darin, mehrere Bilder für verschiedene Endgeräte definieren zu können. Dies soll innerhalb eines `<picture>`-Tags geschehen und kann wie folgt aussehen.<sup>89</sup>

---

89 Vgl. W3 Consortium: Picture Element Proposal,  
[http://www.w3.org/community/respimg/wiki/Picture\\_Element\\_Proposal#Use\\_Cases](http://www.w3.org/community/respimg/wiki/Picture_Element_Proposal#Use_Cases) (29.10.2012)

```
<picture alt="beispiel">
  <source src="bild-mobil.jpg">
  <source src="bild-medium.jpg" media="min-width: 480px">
  <source src="bild-gross.jpg" media="min-width: 800px">
  
</picture>
```

#### 4.5.4 Umsetzung einer Navigation

Nachfolgend soll beschrieben werden, wie ein Layoutwechsel einer Navigation hin zu einem ausklappbaren Menu realisiert werden kann auf Basis von *jQuery*. Grundlage wird dabei eine einfache `<ul>`-Liste:

```
<nav id="navigation">
  <ul id="menu">
    <li><a href="#">Link</a></li>
    <li><a href="#">Link</a></li>
    <li><a href="#">Link</a></li>
  </ul>
</nav>
```

Die Liste erhält einige grundlegende CSS-Eigenschaften, auf Gestaltung wird der Übersicht halber verzichtet:

```
ul#menu li a {
  display: block;
  padding: 10px;
  float: left;
}
```

Als nächsten Schritt benutzen wir *jQuery*, um ab einer bestimmten Browserbreite anstelle der Liste ein Ausklappmenu vorzubereiten, welches nachfolgend durch weiteres CSS gestyled wird. Als erstes wird `#menu` dazu eine neues Element vorangestellt,

welches als Button dienen soll, um das Menu auszuklappen. Dieser erhält die ID ausklapp-button. Anschließend definieren wir die Funktion, welche das Verhalten des Buttons (das Ein- und Ausfahren des Menus) festlegt.

```
<script type="text/javascript" src="http://ajax.googleapis.com/ajax/
libs/jquery/jquery.min.js"></script>
<script type="text/javascript">
jQuery(document).ready(function($){
    $('#navigation').prepend('<div id="ausklapp-button">Menu</div>');
    $('#ausklapp-button').on("click", function(){
        $('#menu').slideToggle();
        $(this).toggleClass("active");
    });
});
</script>
```

Der Browser wird den HTML-Code dank des Scripts wiefolgt rendern:

```
<nav id="navigation">
    <div id="ausklapp-button">Menu</div>
    <ul id="menu">
        <li><a href="#">Link</a></li>
        <li><a href="#">Link</a></li>
        <li><a href="#">Link</a></li>
    </ul>
</nav>
```

Zuletzt kommen Media Queries zum Einsatz. In unserem Beispiel soll das Menu unter eine Browserbreite von 480 Pixeln zu einem Ausklapp-Menu wechseln. Anderenfalls soll der Ausklapp-Button ausgeblendet und das Menu wie bereits festgelegt aussehen.

```
@media screen and (min-width: 481px) {
    #navigation {
        display: block !important;
```



```
    }  
    #ausklapp-button {  
        display: none;  
    }  
    ul#menu li a {  
        display: block;  
        padding: 10px;  
        float: left;  
    }  
}  
@media screen and (max-width: 481px) {  
    #ausklapp-button {  
        display: block;  
        padding: 10px;  
    }  
    #navigation {  
        position: absolute;  
        z-index: 1000;  
        display: none;  
    }  
}
```

Nach entsprechender Gestaltung und Positionierung sieht das Ergebnis im Desktop-Browser und auf mobilen Geräten wie folgt aus:

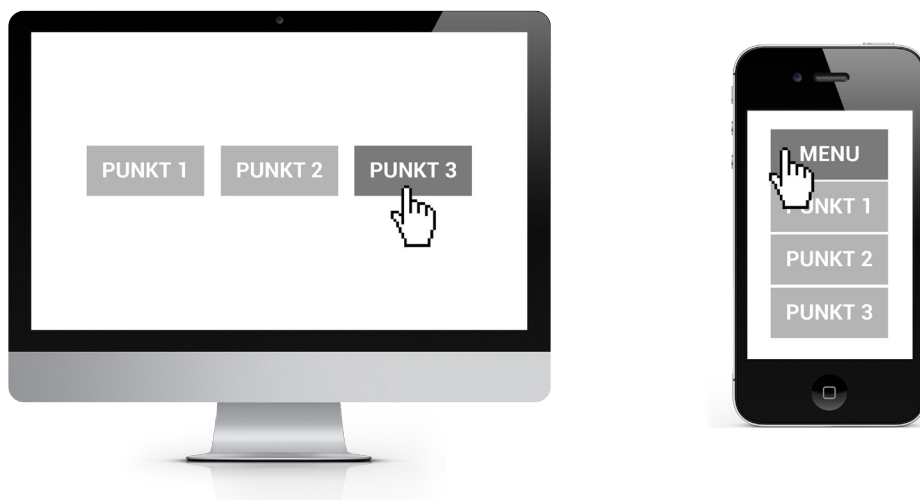


Abb. 30: Vergleich Beispiel-Navigation Desktop-Monitor vs. Smartphone Monitor

### 4.5.6 Gestaltung des Inhalts

Überschriften und andere Text-Elemente müssen beim *Responsive Webdesign* ebenso überarbeitet werden. Auch hier hat sich die Verwendung prozentualer Größen bewährt. Überschriften in einer bestimmten Größe sehen nicht bei jeder Spaltengröße harmonisch aus und müssen entsprechend herunterskaliert werden. Gleiches gilt für Zeilenabstände im Fließtext. Ein nützliches Hilfsmittel ist die Eigenschaft `word-wrap: break-word;`, mit der lange Zeichenfolgen wie URLs bei kleineren Spalten umgebrochen werden.

Zur flexiblen Gestaltung des Textes gibt es einige interessante JavaScript-Plugins, die an dieser Stelle Erwähnung finden sollen. So bringt *FitText* (<http://fittextjs.com/>) große Überschrift dazu, entsprechend der Spaltenbreite dynamisch mitzuskalieren. *On Widows* (<http://artequalswork.com/posts/on-widows.php>) formatiert Fließtext so, dass die letzte Zeile nie aus einem einzigen Wort oder nur wenigen Zeilen besteht. Mittels *Molten leading* (<http://nicewebtype.com/notes/2012/02/03/molten-leading-or-fluid-line-height/>) lassen sich Attribute wie `font-size`, `line-height` and `line-length` dynamisch an die Containerbreite anpassen.

Um Inhalte zu komprimieren und auf das Wichtigste zu konzentrieren, kann das Ausblenden von Inhalten via `display:none`; nützlich sein. Jedes Detail der Hauptseite auch auf der mobilen Version anzeigen zu wollen, muss nicht immer die optimale Lösung sein. Das Ausblenden erfolgt wieder in Verbindung mit entsprechenden Media Queries. Zu beachten ist, dass versteckte Elemente trotzdem vom Browser geladen werden. Die Verwendung von `visibility:hidden`; ist in diesem Fall unpraktisch, da die CSS-Regel das Element dann zwar auch ausblendet, jedoch immer noch den Platz für dieses reserviert.

## 4.6 Responsive Webdesign vs. separate mobile Webseite

Beide Alternativen für plattformunabhängiges Webdesign wurden detailliert vorgestellt und sollen nun auf ihre Vor- und Nachteile untersucht werden. Daraus resultierend werden optimale Einsatzgebiete der beiden Techniken abgeleitet. Um zu

entscheiden, welche Methode für das eigene Webprojekt die passende ist, sollte man folgende Faktoren untersuchen:

*Der Kontext* - Die Bedeutung dieses Faktors wurde bereits bei der Konzeption responsiver Seiten herausgearbeitet. Mit einer separaten mobilen Webseite hat man mehr Gestaltungsmöglichkeiten bezüglich der Inhalte. Man kann die mobile Seite von Grund auf neu entwerfen und sich die Inhalte und Funktionen, die angezeigt werden sollen, gezielt aussuchen. Diese Freiheiten sind bei adaptiven Layouts begrenzt, da man als Basis die vollständige Seite hat und diese nur neu gestalten und anordnen kann. Sollte es hier keine großen Unterschiede geben, ist ein solches Layout die effektivere Lösung. Jedoch ist auch zu bedenken, dass die auf einer separaten mobilen Webseite ausgewählten Inhalte keinesfalls den Erwartungen der Besucher entsprechen müssen. Eine Studie des Unternehmens *Tealeaf* ergab, dass 85 Prozent von einer mobilen Webseite mindestens dasselbe erwarten wie von der Desktop-Version.<sup>90</sup> Für spezielle Dienste wie beispielsweise *Wetter.de* macht eine separate mobile Webseite, die sich nur auf den angebotenen Dienst der Wettervorhersage beschränkt, jedoch Sinn.

*Der Content* - Ein weiterer mit diesem Faktor verbundener Aspekt sind schnellstmögliche Ladezeiten. Dies lässt sich mit einer separaten mobilen Webseite, bei der nur ausgewählte Inhalte auftauchen, leichter realisieren als bei adaptiven Layouts, wo alle Inhalte inklusive Bildern heruntergeladen werden müssen. Selbst Inhalte, die via CSS ausgeblendet werden und somit gar nicht auftauchen, müssen vom Browser vollständig heruntergeladen werden.

*Die Rahmenbedingungen* - Bietet sich ein adaptives Layout aufgrund der vorherigen Faktoren an, kann eine separate mobile Webseite trotzdem die bessere Alternative sein. Denn ist beispielsweise bereits eine angemessene Webpräsenz vorhanden, wäre der Aufwand, der mit der Erstellung einer separaten Webseite verbunden ist, geringer und möglicherweise vorzuziehen. Denn ein adaptives Layout würde bedeuten, die gesamte Webseite neu zu entwerfen. Auch die entstehenden Kosten sind dabei zu beach-

---

90 Vgl. *Tealeaf: New Mobile Transaction Research 2011*,  
<http://www.tealeaf.com/news/news-releases/2011/Tealeaf-Announces-New-Mobile-Research.php>  
(29.10.2012)

ten. Wird eine Webseite jedoch von Grund auf neu entwickelt, sind vor allem wieder Kontext und Content relevant.

In der Praxis benutzen viele große Portale separate mobile Webseiten. Oftmals weichen diese dann auch stark von der eigentlichen Seite ab. Viele News-Seiten wie *Spiegel Online*, Shops wie *Amazon*, Film-Portale wie *Youtube* oder auch soziale Netzwerke wie *Facebook* nutzen diese Methode. Responsive Webdesign hat sich hingegen auf Seiten, wo keine separaten Inhalte notwendig sind, etabliert. So nutzen viele Firmenwebseiten, Blogs oder auch Künstler-Portfolios adaptive Layouts, um ihre Inhalte zu präsentieren.

## 5 Zusammenfassung und Fazit der Arbeit

Als Fazit dieser Arbeit kann gesagt werden, dass sich die Webgemeinde auch weiterhin in einem Anpassungs- und Entwicklungsprozess befinden wird. Die Entwicklung der multimedialen, mobilen Gesellschaft wird weiter voranschreiten und neue internetfähige Medien werden entstehen. Webentwickler werden dann wieder vor der Aufgabe stehen, Webseiten für diese neuen Geräte zu optimieren.

Derzeit sind es Smartphones und Tablets, die unsere Gesellschaft prägen und unser Verhalten im Internet verändern. Die mobilen Geräte verzeichnen reißende Absätze und es zeigt sich, dass deren Besitzer damit immer häufiger im Internet surfen. Das Web reagiert mit der Weiterentwicklung moderner Webstandards wie HTML5 und CSS3 auf diese Veränderungen und optimiert diese Technologien zunehmend unter dem plattformübergreifenden Aspekten.

Die neuen Webstandards dienen dabei als Grundlage und Werkzeug, flexiblere Webseiten zu entwickeln, die den verschiedenen Anforderungen gewachsen sind. *Responsive Webdesign* ist einer dieser Ansätze, der eine Antwort auf dieses Problem gefunden hat. Viele Aspekte der Methode sind derzeit noch nicht optimal gelöst und die Technik macht in jedem Fall Sinn, jedoch erfüllt sie oft auch ihren Zweck mit bravour. In anderen Situationen haben sich dafür separate mobile Webseiten bewährt.

In beiden Fällen kann jedoch nicht von einer endgültigen Lösung gesprochen werden. Die nahe Zukunft ist abzusehen, da die WHATWG Arbeitsgruppe die Webstandards im Hinblick auf eine immer besser zu realisierende plattformunabhängige Webentwicklung kontinuierlich ergänzt und verbessert. Diese Arbeit hat den *aktuellen* Stand der Technik jedoch detailliert analysiert und aufgezeigt, was modernes und plattformunabhängiges Webdesign im Jahr 2012 bedeutet.

# Glossar

API	API (Application Programming Interface) bezeichnet eine Schnittstelle zu einer in sich abgeschlossenen Software, die Entwicklern Funktionen zur Verfügung stellt, durch die Teile des Programms gesteuert werden können.
Button	Button, zu Deutsch Schaltfläche, ist eine Fläche auf einer Webseite oder in einer Anwendung, die durch das Anklicken mit der linken Maustaste eine Aktion auslöst.
Browser	Ein Browser ist die umgangssprachliche Form von Webbrowser. Ein Browser ist eine Software, die Internetseiten aufrufen und entsprechend anzeigen kann.
CSS3	CSS ist die Abkürzung für Cascading Style Sheets und ist mittlerweile schon in der dritten Version in vielen Browsern einsetzbar. Die Stylesheet-Datei beinhaltet rein beschreibende Informationen, wie Elemente/Tags auf der dazugehörigen Webseite angezeigt werden.
DocType	Der DocType, zu Deutsch Dokumententyp, definiert die Art eines Webdokumentes.
Domain	Domain, zu Deutsch Domäne, ist die URL einer Website. Eine Domäne ist aus mindestens 2 Teilen zusammengesetzt, dem Namen (z.B. cross-apps) und der Endung (z.B. com) getrennt durch einen Punkt.
DropDown-Menu	Ein spezielles Menu im Webdesign, welches sich durch seine dynamische Funktionsweise auszeichnet, einige Menüpunkte werden meist erst durch einen RollOver-Effekt angezeigt
Fallback-Strategie	Bezeichnet die Einrichtung von Techniken, um nicht unterstützte Web-Funktionen auch auf älteren Browsern sichtbar zu machen.
Feature-Phone	Bezeichnet eine Kategorie von Mobiltelefonen, die aufgrund des Mangels bestimmter Funktionen nicht als Smartphones bezeichnet werden können.
Header	Header, zu Deutsch Kopfbereich, bezeichnet den oberen

---

	Teil einer Webseite, in dem auf Designebene meistens das Logo in Form eines Banners zu sehen ist.
JavaScript	JavaScript ist eine Skriptsprache, die hauptsächlich Web-Browsern eingesetzt wird.
JavaScript-Bibliothek	Eine JavaScript-Bibliothek ist eine Code-Sammlung, die das Entwickeln JavaScript-basierter Anwendungen erleichtert.
jQuery	Eine populäre JavaScript-Bibliothek.
Plugin	Ein Plug-In ist eine Softwarekomponente, die in eine bestehende Software eingesetzt werden kann.
Slicing	Slicing bezeichnet im Webdesign eine Technik, bei der mehrere für das Layout relevante Grafikelemente in einer Bilddatei zusammengefasst werden.
Smartphone	Ein Smartphone ist ein Mobiltelefon, auf dem sich Apps installieren lassen und das einen Internetzugang besitzt.
Tablet	Tablet-PCs ist ein tragbarer, flacher Computer in besonders leichter Ausführung mit einem Touchscreen-Display.
Tag	Ein Tag, zu Deutsch Markierung, wird im Zusammenhang mit HTML zur Kennzeichnung eines Elementes benutzt.
URL	URL (Uniform Resource Locator) ist die eindeutige Adresse eines Dokumentes im Internet.
Usability	Im Zusammenhang mit Webdesign bezeichnet Usability die Benutzerfreundlichkeit einer Webseite.
Webseite	Sammlung von Homepage und allen Unterseiten, die zu einer Website zusammengefasst werden.

## Quellenverzeichnis

Amazon	Amazon Kindle technische Details, <a href="http://www.amazon.de/Kindle-Zoll-Ink-Display-WLAN-Schwarz/dp/B007HCCOD0/ref=sr_tr_sr_1?ie=UTF8&amp;qid=1351501107&amp;sr=8-1">http://www.amazon.de/Kindle-Zoll-Ink-Display-WLAN-Schwarz/dp/B007HCCOD0/ref=sr_tr_sr_1?ie=UTF8&amp;qid=1351501107&amp;sr=8-1</a> (29.10.2012)
AGOF	Mobile Facts 2011, Seite 14
APPRUPT GmbH	App-Nutzung und Kriterien für den App-Download, Seite 2
androidnext.de	LG: Über-Retina-Display mit 1080p auf 5 Zoll und 440 ppi gezeigt, 28.05.2012, <a href="http://www.androidnext.de/news/lg-display-1080p-5-zoll-440-ppi/">http://www.androidnext.de/news/lg-display-1080p-5-zoll-440-ppi/</a> (29.10.2012)
Atkings, Tab Jr.	There is no such thing as CSS4, 05.09.2012, <a href="http://www.xanthir.com/b4Ko0">http://www.xanthir.com/b4Ko0</a> (29.10.2012)
Bacon, Gary	Responsive Design for Retina Displays on iPad and iPhone, 16.05.2012, <a href="http://garybacon.com/post/responsive-design-for-retina-displays-on-ipad-and-iphone/">http://garybacon.com/post/responsive-design-for-retina-displays-on-ipad-and-iphone/</a> (29.10.2012)
binvisions	List of Tablet and Smartphone Resolutions and Screen Sizes, <a href="http://www.binvisions.com/articles/tablet-smartphone-resolutions-screen-size-list/">http://www.binvisions.com/articles/tablet-smartphone-resolutions-screen-size-list/</a> (29.10.2012)
BITKOM	Smartphone-Absatz steigt rasant, 09.01.2012, <a href="http://www.bitkom.org/de/markt_statistik/64086_70921.aspx">http://www.bitkom.org/de/markt_statistik/64086_70921.aspx</a> (29.10.2012)



Blast Advanced Media	Responsive Web Design Increases Sales, <a href="http://www.blastam.com/assets/pdf/blast-thinktank-photo-responsive-casestudy.pdf">http://www.blastam.com/assets/pdf/blast-thinktank-photo-responsive-casestudy.pdf</a> (29.10.2012)
Bloomberg Businessweek	Before iPhone and Android Came Simon, 29.06.2012, <a href="http://www.businessweek.com/articles/2012-06-29/before-iphone-and-android-came-simon-the-first-smartphone">http://www.businessweek.com/articles/2012-06-29/before-iphone-and-android-came-simon-the-first-smartphone</a> (29.10.2012)
builtwith.com	HTML5 DocType Usage Statistics, <a href="http://trends.builtwith.com/docinfo/HTML5-DocType">http://trends.builtwith.com/docinfo/HTML5-DocType</a> (29.10.2012)
Canalys	Worldwide smart phone market, full year 2011, 03.02.2012, <a href="http://www.canalys.com/news-room/smart-phones-overtake-client-pcs-2011">http://www.canalys.com/news-room/smart-phones-overtake-client-pcs-2011</a> (29.10.2012)
channelpartner.de	Absatzzahlen von Tablet-PCs weiterhin ein Trauerspiel, 10.11.2003, <a href="http://www.channelpartner.de/news/209592/index.html">http://www.channelpartner.de/news/209592/index.html</a> (29.10.2012)
Comic Sans Criminal	You're a Comic Sans Criminal, <a href="http://comics-anscriminal.com/">http://comics-anscriminal.com/</a> (29.10.2012)
Computer History Museum	The 40th Anniversary of the Dynabook, <a href="http://www.computerhistory.org/collections/accession/102695024">http://www.computerhistory.org/collections/accession/102695024</a> (29.10.2012)
Fontspring	Further Hardening of the Bulletproof Syntax, <a href="http://www.fontspring.com/blog/further-hardening-of-the-bulletproof-syntax">http://www.fontspring.com/blog/further-hardening-of-the-bulletproof-syntax</a> (29.10.2012)

- Fontsquirl  
font-face Kits, <http://www.fontsquirl.com/fontface> (29.10.2012)
- Gartner  
Worldwide Media Tablet Sales 2010, 15.10.2010, <http://www.gartner.com/it/page.jsp?id=1452614> (29.10.2012)  
  
World-Wide Media Tablet Sales to End Users, 15.10.2011, <http://www.gartner.com/it/page.jsp?id=1452614> (29.10.2012)  
  
World-Wide Mobile Terminal Sales to End Users in 1Q11, 19.05.2011, <http://www.gartner.com/it/page.jsp?id=1689814> (29.10.2012)  
  
World-Wide Sales of Media Tablets to End Users by OS, 10.04.2012, <http://www.gartner.com/it/page.jsp?id=1980115> (29.10.2012)  
  
World-Wide Smartphone Sales, [http://en.wikipedia.org/wiki/File:World\\_Wide\\_Smartphone\\_Sales\\_Share.png](http://en.wikipedia.org/wiki/File:World_Wide_Smartphone_Sales_Share.png) (29.10.2012)
- GIZMODO  
The Story Behind Apple's Newton, 19.01.2010, <http://gizmodo.com/5452193/the-story-behind-apples-newton> (29.10.2012)
- Haustein, Stefan u. a.  
Quake II GWT Port, <http://code.google.com/p/quake2-gwt-port/> (29.10.2012)
- HTC  
T-Mobile Unveils the T-Mobile G1 - the First Phone Powered by Android, 23.09.2012, <http://web.archive.org/web/20110712230204/http://www.htc.com/www/press>.

---

	<code>aspx?id=66338&amp;lang=1033</code> (29.10.2012)
IDC	Top Five Worldwide Smartphone Vendors, 06.02.2012, <a href="http://www.idc.com/getdoc.jsp?containerId=prUS23299912">http://www.idc.com/getdoc.jsp?containerId=prUS23299912</a> (29.10.2012)  Top Smartphone Operating Systems, 08.08.2012, <a href="http://www.idc.com/getdoc.jsp?containerId=prUS23638712">http://www.idc.com/getdoc.jsp?containerId=prUS23638712</a> (29.10.2012)  Worldwide Media Tablet Shipments, Second Quarter 2012, 02.08.2012, <a href="http://www.idc.com/getdoc.jsp?containerId=prUS23632512">http://www.idc.com/getdoc.jsp?containerId=prUS23632512</a> (29.10.2012)
jQuery	jQuery v1.8.2 <a href="http://code.jquery.com/jquery-1.8.2.min.js">jquery.com</a> , <a href="http://code.jquery.com/jquery-1.8.2.min.js">http://code.jquery.com/jquery-1.8.2.min.js</a> (29.10.2012)
Knight, Kayla	Responsive Web Design: What It Is and How To Use It, 12.02.2011, <a href="http://coding.smashingmagazine.com/2011/01/12/guidelines-for-responsive-web-design/">http://coding.smashingmagazine.com/2011/01/12/guidelines-for-responsive-web-design/</a> (29.10.2012)
Kröner, Peter	HTML5 - Webseiten innovativ und zukunftssicher
Lee, Tim Berners	Reinventing HTML, 27.06.2006, <a href="http://dig.csail.mit.edu/breadcrumbs/node/166">http://dig.csail.mit.edu/breadcrumbs/node/166</a> (29.10.2012)
Lie, Håkon Wium	Cascading Style Sheets, 2005, <a href="http://people.opera.com/howcome/2006/phd/#ch-introduction">http://people.opera.com/howcome/2006/phd/#ch-introduction</a> (29.10.2012)

---

Marcotte, Ethan	Resonsive Webdesign, 25.05.2010, <a href="http://www.alistapart.com/articles/responsive-web-design/">http://www.alistapart.com/articles/responsive-web-design/</a> (29.10.2012)
Microsoft	Internet Explorer 10 - 3D Transforms, <a href="http://msdn.microsoft.com/en-us/library/ie/hh673529(v=vs.85).aspx">http://msdn.microsoft.com/en-us/library/ie/hh673529(v=vs.85).aspx</a> (29.10.2012)
Modernizr	Modernizr Features, <a href="http://modernizr.com/docs/#features-html5">http://modernizr.com/docs/#features-html5</a> (29.10.2012)
Nintendo	Nintento DS Browser, <a href="http://www.nintendo.de/NOE/de_DE/games/nds/nintendo_ds_browser_3151.html">http://www.nintendo.de/NOE/de_DE/games/nds/nintendo_ds_browser_3151.html</a> (29.10.2012)
NPD Group	Top U.S. Smartphone Manufacturers, 02.05.2012, <a href="https://www.npd.com/wps/portal/npd/us/news/press-releases/pr_120502/">https://www.npd.com/wps/portal/npd/us/news/press-releases/pr_120502/</a> (29.10.2012)
nokiamuseum.com	Nokia 9000, <a href="http://nokiamuseum.com/view.php?model=9000">http://nokiamuseum.com/view.php?model=9000</a> (29.10.2012)
Orange World	Mobile shopping is popular when watching TV, says Orange UK research, 16.12.2010, <a href="http://thefonecast.com/Home/tabid/61/ArticleID/3602/ArtMID/538/Default.aspx">http://thefonecast.com/Home/tabid/61/ArticleID/3602/ArtMID/538/Default.aspx</a> (29.10.2012)
Pew Research Center's for Excellence in Journalism	The Explosion in Mobile Audiences And A Close Look At What It Means For News, 01.10.2012, <a href="http://www.journalism.org/analysis_report/future_mobile_news">http://www.journalism.org/analysis_report/future_mobile_news</a> (29.10.2012)

Prismo	Desktop LCD Display Comparison, <a href="http://www.prismo.ch/comparisons/desktop.php">http://www.prismo.ch/comparisons/desktop.php</a> (29.10.2012)
Quirksmode	CSS contents and browser compatibility, <a href="http://www.quirksmode.org/css/contents.html">http://www.quirksmode.org/css/contents.html</a> (29.10.2012)
Samsung	Anwendungsmöglichkeiten von Samsung Smart TV, <a href="http://de.samsung.com/de/microsites/smarttv/anwendungen_feature.aspx">http://de.samsung.com/de/microsites/smarttv/anwendungen_feature.aspx</a> (29.10.2012)  Samsung Galaxy Note 10.1, <a href="http://www.samsung.com/de/consumer/mobile-device/galaxy-note/galaxy-note/GT-N8010ZWADBT">http://www.samsung.com/de/consumer/mobile-device/galaxy-note/galaxy-note/GT-N8010ZWADBT</a> (29.10.2012)
Shaw, Brad	Why are CSS3 transitions worth using?, <a href="http://css3.bradshawenterprises.com/blog/jquery-vs-css3-transitions/">http://css3.bradshawenterprises.com/blog/jquery-vs-css3-transitions/</a> (29.10.2012)
Statistika GmbH	Smartphone-Nutzer in Deutschland von 2009 bis 2012, <a href="http://de.statista.com/statistik/daten/studie/198959/umfrage/anzahl-der-smartphonenutzer-in-deutschland-seit-2010/">http://de.statista.com/statistik/daten/studie/198959/umfrage/anzahl-der-smartphonenutzer-in-deutschland-seit-2010/</a> (29.10.2012)  Global market share of Nokia smartphones since 2012, <a href="http://www.statista.com/statistics/12861/market-share-of-nokia-smartphones-since-2007/">http://www.statista.com/statistics/12861/market-share-of-nokia-smartphones-since-2007/</a> (29.10.2012)  Smartphone-Funktionen - Nutzung 2012, <a href="http://de.statista.com/statistik/daten/studie/166150/umfrage/nutzung-von-smartphone-funktionen-">http://de.statista.com/statistik/daten/studie/166150/umfrage/nutzung-von-smartphone-funktionen-</a>

---

in-deutschland/ (29.10.2012)

Tealeaf

New Mobile Transaction Research 2011, <http://www.tealeaf.com/news/news-releases/2011/Tealeaf-Announces-New-Mobile-Research.php> (29.10.2012)

W3 Consortium

<audio> Element, <http://www.w3.org/wiki/HTML/Elements/audio> (29.10.2012)

<video> Element, <http://www.w3.org/wiki/HTML/Elements/video> (29.10.2012)

3D Transfrom Properties, 16.07.2009, <http://www.w3.org/TR/css3-3d-transforms/#transform-property> (29.10.2012)

Cascading Style Sheets, level 1, 17 Dec 1996, <http://www.w3.org/TR/REC-CSS1/> (29.10.2012)

CSS Backgrounds and Borders Module Level 3, <http://www.w3.org/TR/css3-background/#the-border-radius> (29.10.2012)

CSS Flexible Box Layout Module, 18.10.2012, <http://www.w3.org/TR/css3-flexbox/> (29.10.2012)

CSS Fonts Module Level 3, 23.08.2012, <http://www.w3.org/TR/css3-fonts/#font-face-rule> (29.10.2012)

SS Image Values and Replaced Content Module Level 3, 12.06.2012, <http://dev.w3.org/csswg/>

css3-images/ (29.10.2012)

CSS Multi-column Layout Module, 12.04.2011,  
<http://www.w3.org/TR/css3-multicol/>  
(29.10.2012)

CSS Text Level 3, 14.08.2012, <http://www.w3.org/TR/css3-text/#text-shadow> (29.10.2012)

CSS Transforms, 11.09.2012, <http://www.w3.org/TR/css3-2d-transforms/#two-dimensional-subset> (29.10.2012)

Inline SVG in HTML5 and XHTML, <http://dev.w3.org/SVG/proposals/svg-html/svg-html-proposal.html> (29.10.2012)

HTML: The Markup Language (an HTML language reference), <http://www.w3.org/TR/html-markup/elements.html> (29.10.2012)

HTML: The Markup Language (an HTML language reference) - article, <http://www.w3.org/TR/html-markup/article.html#article> (29.10.2012)

HTML: The Markup Language (an HTML language reference) - aside, <http://www.w3.org/TR/html-markup/aside.html#aside> (29.10.2012)

HTML: The Markup Language (an HTML language reference) - hgroup, <http://www.w3.org/TR/html-markup/hgroup.html#hgroup> (29.10.2012)

HTML: The Markup Language (an HTML language

ge reference) - section, <http://www.w3.org/TR/html-markup/section.html#section> (29.10.2012)

Media Queries, 19.06.2012, <http://www.w3.org/TR/css3-mediaqueries/> (29.10.2012)

Media Types, <http://www.w3.org/TR/2002/WD-CSS21-20020802/media.html> (29.10.2012)

Mobile Web Application Best Practices, 14.12.2010, <http://www.w3.org/TR/mwabp/#bp-viewport> (29.10.2012)

Selectors Level 3, 29.09.2011, <http://www.w3.org/TR/selectors/> (29.10.2012)

Web Storage, <http://www.w3.org/TR/webstorage/> (29.10.2012)

#### w3schools

Browser Display Statistics, [http://www.w3schools.com/browsers/browsers\\_display.asp](http://www.w3schools.com/browsers/browsers_display.asp) (29.10.2012)

OS Platform Statistics, [http://www.w3schools.com/browsers/browsers\\_os.asp](http://www.w3schools.com/browsers/browsers_os.asp) (29.10.2012)

#### Webhits

Web-Barometer, <http://www.webhits.de/deutsch/index.shtml?webstats.html> (29.10.2012)

#### WebKit

3D Transforms, 16.07.2009, <https://www.webkit.org/blog/386/3d-transforms/> (29.10.2012)

#### Webtrekk

Webtrekk Deutsche Webstatistik 3. Quartal 2012, 02.10.2012, <http://www.webtrekk.com/>



fileadmin/pdf/pm/PM\_2012/Webtrekk\_  
Studie\_2012-Q3\_DE.pdf (29.10.2012)

Willison , Simon

Rounded Corners with CSS and JavaScript,  
28.05.2004, <http://www.sitepoint.com/rounded-corners-css-javascript/> (29.10.2012)

Wilson, Tracy V.

How the iPhone Works, <http://electronics.howstuffworks.com/iphone5.htm> (29.10.2012)

Wroblewski, Luke

Mobile First, 19.01.2012, <http://www.abookapart.com/products/mobile-first> (29.10.2012)

## Eigenständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe. Stellen, die wörtlich oder sinngemäß aus Quellen entnommen wurden, sind als solche kenntlich gemacht. Diese Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt.

---

Ort

---

Datum

---

Nachname, Vorname

---

Unterschrift

